

1989

GC
7.1
D 43
1988

A Control System Design Technique for Nonlinear Discrete Time Systems

by

David Matthew DeLonga

B.S.M.E., United States Naval Academy (1980)
S.M.M.E., Massachusetts Institute of Technology (1981)

Submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

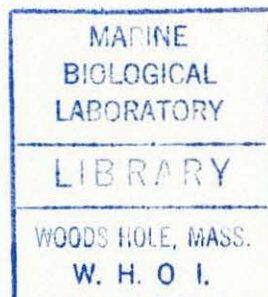
and the

WOODS HOLE OCEANOGRAPHIC INSTITUTION

February 1989

© David M. DeLonga, 1988

The author hereby grants to MIT permission to reproduce and
to distribute copies of this thesis document in whole or in part.



Signature of Author

Joint Program in Oceanographic Engineering
Massachusetts Institute of Technology
Woods Hole Oceanographic Institution
September 10, 1988

Certified by

Dana R. Yoerger
Associate Scientist, Woods Hole Oceanographic Institution
Thesis Supervisor

Certified by

Jean-Jacques E. Slotine
Assistant Professor, Massachusetts Institute of Technology
Thesis Supervisor

Accepted by

W. Kendall Melville
Chairman, Joint Committee for Oceanographic Engineering
Massachusetts Institute of Technology/Woods Hole Oceanographic Institution

Gift

W.H.O.I.

A Control System Design Technique for Nonlinear Discrete Time Systems

by

David Matthew DeLonga

Submitted to the Massachusetts Institute of Technology—
Woods Hole Oceanographic Institution
Joint Program in Oceanographic Engineering
on September 10, 1988, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

A new control methodology is proposed for use with a class of nonlinear, single-input discrete time systems. The technique is based on a discrete time approach that parallels existing continuous time sliding surface concepts. Modifications to the basic algorithm allow for system models with time-variant or uncertain parameters, time delays in the control input, and external disturbances. A major feature of the method is its straightforward extension to an adaptive control form which can be used to improve performance and maintain stability in the presence of large parametric uncertainty or time-variant behavior. Techniques are proposed for overcoming instabilities that frequently arise when using adaptive control schemes based on reduced order system models or in the presence of disturbances.

A framework is provided for the practical application of the methodology to continuous time systems. The discrete time nature of the development makes it especially well suited to applications where sensor data is infrequently available or computational power is limited. An experimental study is performed using an underwater remotely operated vehicle to verify the validity of the approach. The ability of the method to use a nonlinear model and adapt to large parametric uncertainty is shown to result in improved performance over the use of a linear or time-invariant model.

Thesis Supervisor: Dana R. Yoerger

Title: Associate Scientist, Woods Hole Oceanographic Institution

Thesis Supervisor: Jean-Jacques E. Slotine

Title: Assistant Professor, Massachusetts Institute of Technology

Acknowledgements

I dedicate this document to my parents who encourage and support me in all my endeavors.

In preparation of this document, I would like to acknowledge my appreciation to the following:

- My advisors, Dana Yoerger and Jean-Jacques Slotine, for their guidance, encouragement, and support.
- Art Baggeroer, for serving on my committee.
- Robert Ballard and the members of the Deep Submergence Laboratory for their assistance and support.
- Hagen Schempf and my fellow students for their constant encouragement and friendship.
- Judy White, for making this thesis a reality.

Finally, the United States Navy and the National Science Foundation are gratefully acknowledged for their financial support of my graduate education. This research was also sponsored in part by ONR Contract N00014-36-C-0038, ONR Grant N00014-87-J-1111 (formerly N00014-87-G-0111), NSF Grant 8611640-ECS, and NRL Contract N00014-88-K-2022.

Contents

Acknowledgements	3
1 Introduction	11
1.1 Motivation	11
1.2 Research Objectives	14
1.2.1 Theoretical Development	14
1.2.2 Practical Implementation	15
1.3 Overview of Existing Theory and Implementation	16
1.3.1 Nonlinear System Control Theory	16
1.3.2 Discrete Time System Control Theory	17
1.3.3 Adaptive System Control Theory	18
1.3.4 Underwater Vehicle Control Implementation	20
1.4 Thesis Contributions	21
1.5 Outline of the Thesis	22
2 Discrete Time Control of Known Dynamic Systems Using Sliding Surfaces	25
2.1 Discrete Time System Stability Using The Invariant Set Principle	27
2.2 Concept of a Discrete Time Sliding Surface	28
2.3 Control Law Formulation	31
2.4 Selection of the Sliding Surface	34
2.4.1 Linear, Time-Invariant Systems	34

2.4.2	Nonlinear, Time-Variant Systems	37
2.5	Extension To Trajectory Following Controller Form	49
2.6	Extension to Input Time Delays	54
3	Discrete Time Control of Uncertain Dynamic Systems Using Sliding Surfaces	58
3.1	Disturbances	60
3.2	Uncertainty in System Dynamics	67
3.3	Uncertainty in Control Gain	79
4	Discrete Time Adaptive Control of Uncertain Dynamic Systems Using Sliding Surfaces	90
4.1	Preliminary Development	92
4.1.1	System Description, Assumptions, and Definitions	92
4.1.2	A Discrete Time Stability Theorem	95
4.2	Parameter Estimation Methods	98
4.2.1	Normalized Gradient Parameter Estimation	98
4.2.2	Least Squares Parameter Estimation	101
4.2.2.1	Normal Least Squares	101
4.2.2.2	Weighted Least Squares	105
4.2.2.3	Forgetting Factor Least Squares	106
4.2.2.4	Covariance Resetting Least Squares	108
4.2.3	Parameter Convergence	110
4.3	Constrained Parameter Estimation	114
4.3.1	Normalized Gradient	114
4.3.2	Least Squares	114
4.4	Parameter Estimation with Bounded Disturbances	118
4.4.1	Normalized Gradient	118
4.4.2	Least Squares	122

4.5	Adaptive Controller Development Without Unmodeled Dynamics	125
4.5.1	No Disturbance Case	125
4.5.1.1.	Normalized Gradient Parameter Estimation	128
4.5.1.2	Least Squares Parameter Estimation	130
4.5.1.3	Parameter Convergence	131
4.5.2	Extension to Bounded Disturbances	140
4.5.2.1	Normalized Gradient Parameter Estimator	140
4.5.2.2	Least Squares Parameter Estimator	144
4.5.3	Extension to Deterministic Actuator Dynamics	150
4.6	Adaptive Control in the Presence of Unmodeled Dynamics	152
4.6.1	Potential Mechanisms for Instability	152
4.6.1.1	Bandwidth Modification	153
4.6.1.2	Corrupted Adaptation Signals	155
4.6.2	Possible Solutions to Instability Mechanisms	156
4.6.2.1	Bandwidth Control during the Adaptation Process . . .	157
4.6.2.2	Relevant Adaptation	161
5	Physical System Implementation	173
5.1	Modeling	173
5.1.1	System Identification	173
5.1.2	Discrete Time Modeling of Continuous Time Systems	174
5.2	Continuous Time Versus Discrete Time Controller Formulation	178
5.3	System Form	181
5.4	Bandwidth Issues	183
6	Experimental Study	186
6.1	Experimental Apparatus	186
6.1.1	Vehicle Description	186
6.1.2	Sensor Systems	189

6.1.3	Telemetry and Tether System	189
6.1.4	Control System Hardware	190
6.2	Control System Software	192
6.3	Vehicle Modeling	196
6.3.1	System Identification	196
6.3.2	Model Imperfections	199
6.4	Experimental Results	201
6.4.1	Non-Adaptive Control Results	203
6.4.2	Adaptive Control Results	210
7	Summary, Conclusions, and Recommendations	221
	References	227

LIST OF FIGURES

- 1.1 Schematic of the deployed ARGO/JASON system
- 2.1 Continuous time sliding mode behavior and discrete time quasi-sliding mode behavior for a two dimensional system
- 2.2 Phase plane trajectory
- 2.3 S trajectory
- 2.4 Phase plane trajectory as time $\rightarrow \infty$
- 2.5 Error phase plane trajectory
- 2.6 Actual (-) and desired (- -) x_1
- 2.7 Actual (-) and desired (- -) x_2
- 3.1 Sliding surface with boundary layer for a two dimensional system
- 3.2 Error phase plane
- 3.3 Control input using smoothed and discontinuous control law
- 3.4 Maximum eigenvalue magnitude of closed loop system
- 3.5 Maximum eigenvalue magnitude of closed loop system for different bandwidths
- 3.6 S (-) and Φ (- -)
- 3.7 Control input
- 3.8 Contour plot of maximum eigenvalue magnitude
- 3.9 S (-) and Φ (- -)
- 3.10 Control input
- 4.1 S trajectory
- 4.2 Parameter error norm
- 4.3 Actual (-) and estimated (- -) parameter 1
- 4.4 Actual (-) and estimated (- -) parameter 2
- 4.5 Minimum eigenvalue of the regressor matrix
- 4.6 S (-) and Φ (- -)
- 4.7 Parameter error norm
- 4.8 S trajectory

4.9	Parameter error norm	22.8
4.10	S (-) and Φ (- -)	22.8
4.11	Control input	22.8
4.12	Parameter error norm	22.8
4.13	Actual (-) and estimated (- -) parameter 1	22.8
4.14	Actual (-) and estimated (- -) parameter 2	22.8
5.1	Block diagram of s dynamics	22.8
6.1	Experimental apparatus	22.8
6.2	Control system software block diagram	
6.3	Coordinate system	
6.4	Steady state performance in the yaw degree of freedom	
6.5	Model validation in the yaw degree of freedom	
6.6	S (-) and Φ (- -)	
6.7	Actual (-) and desired(- -) x_1	
6.8	Actual (-) and desired(- -) x_2	
6.9	Control input	
6.10	S (-) and Φ (- -)	
6.11	Control input	
6.12	S (-) and Φ (- -) using a linear model	
6.13	Actual (-) and desired(- -) x_1 using a linear model	
6.14	Actual (-) and desired(- -) x_2 using a linear model	
6.15	Controller performance using linear (-) and nonlinear (- -) models	
6.16	S (-) and Φ (- -)	
6.17	Control input	
6.18	S (-) and Φ (- -)	
6.19	Control input	
6.20	Actual (-) and desired(- -) x_2	
6.21	Actual (-) and estimated (- -) parameter 1	

- 6.22 Actual (-) and estimated (- -) parameter 2
- 6.23 Parameter error norm
- 6.24 Actual (-) and estimated (- -) parameter 1
- 6.25 Actual (-) and estimated (- -) parameter 2
- 6.26 S (-) and Φ (- -)
- 6.27 Actual (-) and desired (- -) x_2
- 6.28 S (-) and Φ (- -)
- 6.29 Actual (-) and desired(- -) x_2

Chapter 1

Introduction

1.1 Motivation

The control system designer is often faced with many challenges when applying the available control theoretic principles to a physical system of interest. The ability to model physical system dynamics has generally exceeded the capability to use such models in control system design. Linear control techniques are normally used in the design based on a model obtained through linearization about a nominal system operating point assuming known and time-invariant parameters. Detailed information about the system dynamics is therefore lost when the system state moves away from the operating point or when parameters are poorly known or time-variant. Time delays, infrequent sensor input, limited computational power, and disturbances are also commonly encountered in practice but difficult to analyze rigorously.

This research work was motivated by the need to develop a control design methodology suitable for application on underwater vehicles. An example of the new generation of remotely operated vehicles currently being developed is the ARGO/JASON (Figure 1.1), a dual vehicle system for use in the scientific community. The multi-degree-of-freedom movement and sophisticated tasks of which these systems are capable demand

an effective low-level control system which can implement the commands from an operator or high-level supervisory controller.

Underwater vehicles can present a difficult control system design problem not well suited to traditional approaches. The dynamics of these vehicles are described by high order, nonlinear, time-variant models for which the parameters may be poorly known [1,2]. Discrete time, infrequent sensor inputs (such as acoustic navigation data), and limited computational capability may impose slow sampling rates and thus a large time interval between each control input. Furthermore, the control inputs dictated by a surface controller may need to be sent over a long transmission cable (tether) through a telemetry system that induces time delays in the closed loop system. Finally, these vehicles operate in a high disturbance environment (currents and tether forces) that is difficult to predict with precision.

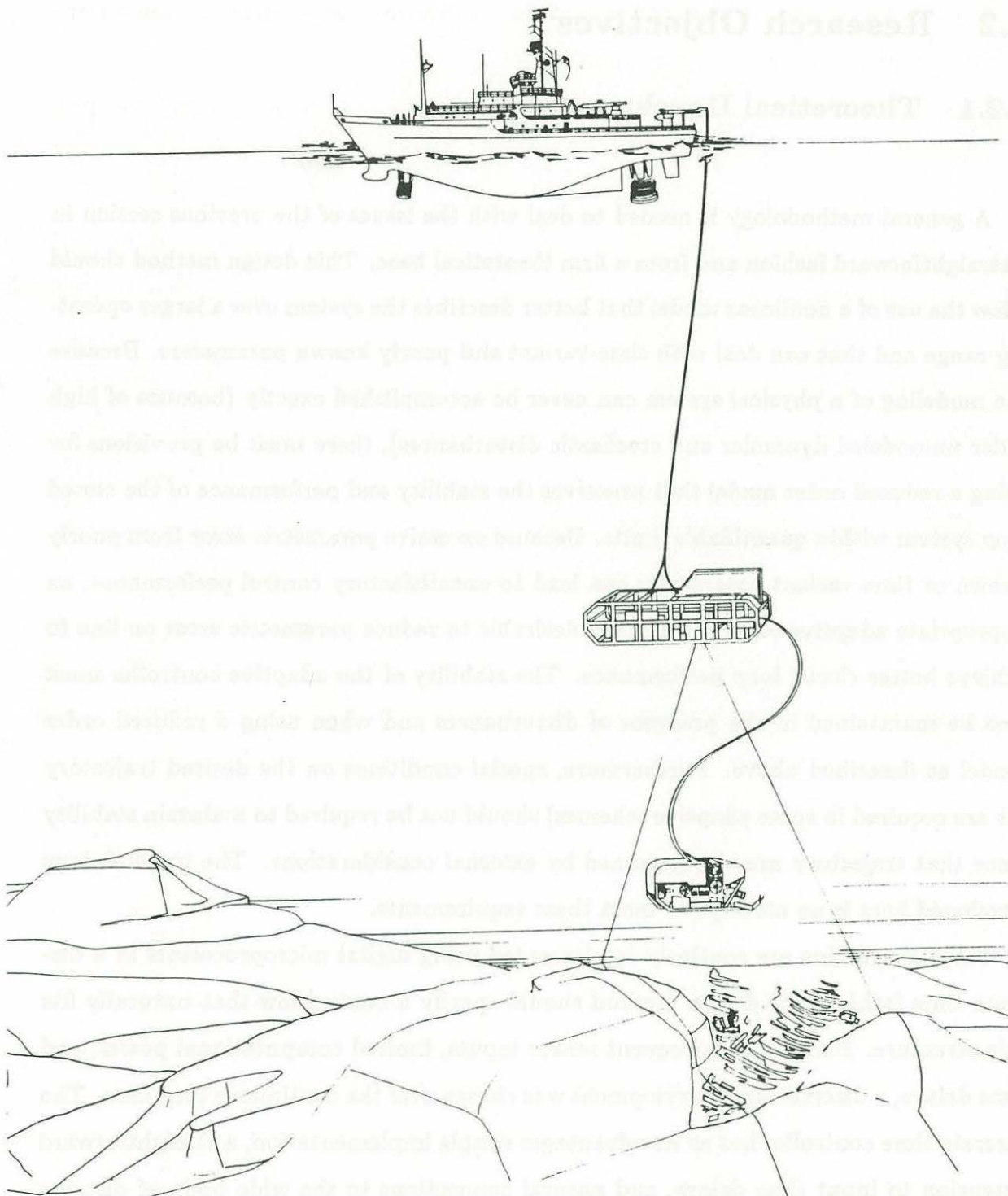


Figure 1.1 Schematic of the deployed ARGO/JASON system

1.2 Research Objectives

1.2.1 Theoretical Development

A general methodology is needed to deal with the issues of the previous section in a straightforward fashion and from a firm theoretical base. This design method should allow the use of a nonlinear model that better describes the system over a larger operating range and that can deal with time-variant and poorly known parameters. Because the modeling of a physical system can never be accomplished exactly (because of high order unmodeled dynamics and stochastic disturbances), there must be provisions for using a reduced order model that preserves the stability and performance of the closed loop system within quantifiable limits. Because excessive parametric error from poorly known or time-variant parameters can lead to unsatisfactory control performance, an appropriate adaptive control feature is desirable to reduce parametric error on line to achieve better closed loop performance. The stability of the adaptive controller must also be maintained in the presence of disturbances and when using a reduced order model as described above. Furthermore, special conditions on the desired trajectory (as are required in some adaptive schemes) should not be required to maintain stability since that trajectory may be governed by external considerations. The methodology developed here is an attempt to meet these requirements.

Since controllers are routinely implemented using digital microprocessors in a discrete time fashion, the design method should specify a control law that naturally fits this structure. Because of infrequent sensor inputs, limited computational power, and time delays, a discrete time development was chosen over the continuous time case. The discrete time controller has as its advantages simple implementation, a straightforward extension to input time delays, and natural connections to the wide body of discrete time estimation theory in the adaptive control case. Disadvantages of the discrete time method include the possible loss of physical insight that occurs when the discrete time

rather than continuous time system equations are utilized, and that nonlinear equations in the continuous time domain are sometimes not able to be expressed exactly in the discrete time domain. The discrete time development can be considered as the more general case in that the algorithms reduce to their continuous time counterparts as the sampling interval approaches zero.

1.2.2 Practical Implementation

Of paramount concern in this research effort was the ability to apply the theoretical concepts developed in a real world situation. As discussed in Section 1.1, the motivation for this research was an application envisioned for an underwater remotely operated vehicle. Because of the limited knowledge of the hydrodynamic effects on these irregularly shaped vehicles, the models representing these vehicles contain numerous imprecisions. It is therefore important that the available theory interface smoothly with these imprecise models.

The practical implementation of theoretically derived concepts also serves as an important source of guidance as to where the present theory is lacking and the direction further research should take. Only by a dedicated implementation effort can the control system designer fully appreciate the limitations of his work and correctly gage its place in the control system design process.

1.3 Overview of Existing Theory and Implementation

This research work covers a rather broad spectrum of specialized topics consisting of nonlinear system control theory, discrete time control theory, adaptive system control theory, and underwater vehicle control implementation. For the purpose of obtaining an overall picture of how this research relates to these areas, a brief overview of these subjects will be provided.

1.3.1 Nonlinear System Control Theory

System control has traditionally focused on the use of linear system descriptions because of the inherent tractability of the resulting equations, both in the time domain and in the frequency domain. Unfortunately, few real life processes can be described exactly by these system descriptions and frequently the approximations made in obtaining a linear description are sufficient to cause a significant degradation in control system performance. There has thus been an interest in extending some of the well tested theoretical work from the linear to the nonlinear case, or for devising new methods which can deal with nonlinear systems.

One possible approach is the use of a modified linear technique such as gain scheduling where a time-variant linear control law is used to account for nonlinearities in the system dynamics. The control gains are "scheduled" or adjusted (based on some signal such as an actual or desired system state variable) in order to maintain a linear, time-invariant system characteristic (e.g. closed loop pole location). A difficulty with this approach is showing that the linear, time-invariant design characteristic is maintained throughout the state space when using the time-variant control law. A heuristic method that is commonly used to simplify the stability analysis relies on utilizing a

scheduling algorithm that results in a relatively slow change in control gains so that quasi-time-invariant arguments can be used. The gain scheduling approach becomes cumbersome if nonlinearities occur in multiple dimensions as the scheduling algorithm expands exponentially with the number of dimensions that require linearization.

A method which has shown promise in the area of directly dealing with system nonlinearities is that of variable structure systems which has received considerable attention in the Soviet Union [3,4]. This method uses a control law which changes structure depending upon the current location of the system state with respect to a "switching hyperplane" in the state space. A recent modification of this work has been devoted toward achieving predictable performance bounds and maintaining robustness to unmodeled dynamics [5]. These methods have concentrated on the continuous time domain and are often motivated by physical system dynamics. Similar work in the discrete time domain [6-8] has also been conducted to account for the digital implementation of modern control systems.

1.3.2 Discrete Time System Control Theory

Discrete time systems theory has attracted increasing interest since the advent of the digital computer. The use of computers to implement control laws is most commonly accomplished using a digital to analog converter which utilizes a "zero-order-hold" device that keeps the control input constant during the sampling interval. The closed loop system can be characterized conveniently if the plant to which the control input is applied can be described in a discrete time fashion. Most continuous time control theoretic principles have been developed in the discrete time domain as well - a recent survey of the current state of the art can be found in [9], and a comprehensive survey of the implementation of digital controllers is also available in [10].

When controlling a physical system with a digital control system, a rigorous discrete time analysis can be done if the continuous time system description can be integrated

exactly over the sampling interval. This is always possible for linear systems, but is generally not possible for nonlinear and/or time-variant systems. Additionally, the mathematical tractability of linear systems has resulted in the majority of discrete time methods being developed for these systems.

The analysis of nonlinear and/or time-variant continuous time systems using a discrete time implementation is therefore made difficult because of the inability to integrate the equations exactly over a given sampling interval in order to obtain a discrete time model. Even if this is possible, a control structure must be available that can adequately deal with the resultant nonlinear discrete time system. Some attempts to deal with the control or modeling of nonlinear, time-variant discrete time systems are presented in [11-13].

1.3.3 Adaptive System Control Theory

Adaptive control has been actively pursued in the last thirty years as a possible solution to the control of poorly known or time-variant systems. The theoretical development in the field was initially limited and resulted in many ad hoc and heuristic rules being developed. The difficulty lies in the fact that the adaptation laws are a function of the system state and thus the resultant closed loop system is nonlinear even if the plant itself is linear. Some surveys of early work and classical references can be found in [14-21].

Renewed interest in the underlying theory in the 1970's resulted in the solving of many issues but with rather rigorous assumptions on the system being controlled. These assumptions have generally included no or limited disturbances and a perfect knowledge of plant order (no unmodeled dynamics). The current methods can be divided into two major categories – direct and indirect. In the direct method, the problem is formulated such that the parameters being adapted on appear directly in the control law. In the indirect method, a separate parameter estimation scheme is

used and combined with a control design procedure to produce the control law. The literature in the adaptive control field is now quite extensive and covers a wide variety of methods [23-35].

Much of the current research is centered on the relaxation of the assumptions listed above in order that the theory be more readily applicable to real world systems. Various approaches have been used to analyze the implication of these relaxations on system stability [36-50]. In fact, some studies have suggested that the current methods will not work even with modest violations of these assumptions. However, it is difficult to predict the effect of disturbances and unmodeled dynamics because of the nonlinearities present – indeed, an approach that utilizes a linear analysis may produce erroneous or misleading results.

Some methods have been proposed for maintaining robustness in adaptive schemes in non-ideal systems. These include a fixed [51-53] and time-variant [54] “deadzone” through which the signal that is driving adaptation is passed, where the size of the deadzone nonlinearity is selected based on the size or frequency range expected in the disturbance environment. The effects of unmodeled dynamics have been reduced by regressor filtering and small adaptation gains [55,56] which permit quasi-linear, time-invariant analysis. Conditions on the input signals of some adaptive systems can sometimes be shown to lead to exponential convergence which tends to reduce the effects of both disturbances and unmodeled dynamics [57-61]. Heuristic rules have also been developed in an effort to eliminate the instabilities [62-65]. There has yet to be a unified approach to this aspect of the problem.

One obstacle confronting the researcher in the adaptive control area is the wide diversity of methods used, each with their own set of conditions and assumptions. For example, linear or nonlinear systems may be specified in either a continuous or discrete time context where a frequency and/or time domain analysis is performed. This sometimes makes it difficult to apply the concepts discovered in one area to those of another method.

1.3.4 Underwater Vehicle Control Implementation

Underwater vehicle control has traditionally been limited by the availability of good sensors as well as by a sparsity of control methods that are viable for a poorly known nonlinear and/or time-variant system operating in a high disturbance environment. Position information is usually obtained by an acoustic tracking system which, by the nature of its operation, provides discrete position fixes with a periodicity dependent on range and sound speed with accuracy dependent on frequency. Velocity information is obtained by strap down sensor packages or by state estimation, with inertial sensors being used in more sophisticated applications. Control inputs can be obtained by water jet or thruster (propeller) action.

Many vehicles have been outfitted with heading or depth regulators that are designed using simple linear controllers with a model obtained by linearization about a stationary operating point. Some more sophisticated controllers have been proposed or implemented [66-68] that are capable of trajectory following (position and velocity) in multi-degree-of-freedom motion. As discussed in Section 1.1, the presence of telemetry time delays and disturbances greatly complicates the control of these vehicles.

1.4 Thesis Contributions

The first contribution of this thesis is the introduction of a control design methodology for a class of nonlinear and time-variant discrete time systems with uncertain parameters and subject to external disturbances. The second contribution is a new method for the adaptive control of single-input, multi-output discrete time systems. A third contribution is a set of techniques for maintaining the stability of an adaptive control system when using reduced order models in higher order systems. The fourth contribution is an implementation of the above methods and an experimental study using an underwater remotely operated vehicle.

1.5 Outline of the Thesis

Chapter 2 presents the discrete time control of known dynamic systems using sliding surfaces. Beginning with a brief discussion of discrete time system stability, the concept of a discrete time sliding mode behavior is introduced and contrasted with its continuous time analog. A control law is then proposed to reach and maintain the discrete time sliding mode behavior from any initial state for single-input discrete time systems with full state feedback. The method for selecting a sliding surface with the desired properties is demonstrated for both linear systems and a class of nonlinear and time-variant systems. The method is next extended to a trajectory following controller form and for an arbitrary delay in the control input.

Chapter 3 concerns the discrete time control of uncertain dynamic systems using sliding surfaces. A subset of the class of systems considered in Chapter 2 is considered where disturbances and errors in system dynamics and control gain constitute the uncertainties present. A discontinuous form of the control law presented in Chapter 2 is shown to drive the system state into a "boundary layer" surrounding the sliding surface. Conditions are placed on the magnitude of the uncertainties such that stability within the boundary layer around the sliding surface is assured. Because this discontinuous control input can potentially excite unmodeled dynamics in the system being controlled, a filtered version of the controller is used when within this boundary layer. Performance expectations can be generated dependent on the magnitude of uncertainty present and the control bandwidth desired.

The discrete time adaptive control of the systems considered in Chapters 2 and 3 is covered in Chapter 4, with the additional restriction that the systems are linear in an unknown parameter vector. The chapter begins with an additional discrete time stability proof that is required for the adaptive controller which has its origin in the discrete time adaptive control of single-input, single-output systems. Parameter estimation methods are then considered along with their convergence properties. Estimation

using a constrained parameter set and in the presence of bounded disturbances is also discussed. The adaptive controller formulation follows and is based on the control laws developed in Chapters 2 and 3. This controller is also extended to the cases of constrained parameter sets and bounded disturbances. It is shown how known unmodeled dynamics which have the form of actuator dynamics can be explicitly accounted for in the parameter estimator and control law. An analysis is then provided for the system behavior when the adaptive controller is used in the presence of unmodeled dynamics. Specifically, two mechanisms are described which can potentially create closed loop system instability in this situation – bandwidth modification and the corruption of signals used in the parameter estimator. Possible solutions are provided which, by controlling bandwidth during the adaptation process and by adapting only when relevant information is present, may overcome these instability mechanisms. These solutions require reasonable assumptions on the nature of the unmodeled dynamics present.

Chapter 5 addresses issues regarding implementation of the methodology on physical systems. Modeling techniques are discussed which can be used to form the requisite discrete time model and an estimate of the model uncertainties. The advantages and disadvantages of using a discrete versus continuous time controller are discussed, and guidance provided as to which is preferable under different circumstances. The applicability of the method to systems which do not fall into the exact form as that dealt with in Chapters 2 through 4 is investigated. Guidance pertaining to bandwidth selection in the discrete time formulation based on characteristics of the continuous time system and model imprecisions is also covered.

Chapter 6 contains an experimental study using an underwater remotely-operated vehicle as the system to be controlled. A description is given of the vehicle and its associated hardware and software. The modeling process is explained and both linear and nonlinear models obtained along with the model imperfections believed to exist. Experimental results are presented both for the non-adaptive and adaptive controller formulations, and a comparison made between the performance obtained using the

linear and nonlinear models. The instability that can occur when using adaptive control in the presence of unmodeled dynamics is shown as well as the success of using the proposed methods to overcome this behavior.

Chapter 7 summarizes the results of the thesis and offers recommendations for future research.

Chapter 2

Discrete Time Control of Known Dynamic Systems Using Sliding Surfaces

In this chapter the use of discrete time sliding surfaces is introduced as a possible control strategy as motivated by continuous time sliding surface control methodologies. A fundamental difference between the two approaches is that in the former method the control input is applied at discrete sampling instants with the system effectively operating in open loop between sampling instants. Known dynamic systems will be considered here, and various classes of uncertainties will be considered in the next chapter.

Discrete time system stability using the invariant set principle is presented initially as it is pivotal in the following controller development. Next the concept of a discrete time sliding surface is discussed and contrasted with the corresponding continuous time case. The control law is then formulated to cause the system state to move along the discrete time sliding surface. The method of selecting an appropriate sliding surface such that the desired stability properties are achieved in the closed loop system is discussed. This selection is possible for general linear, time-invariant systems and for

a limited class of nonlinear, time-variant systems. The methodology is extended to arbitrary time delays in the control input, where these delays are an integral multiple of the sampling interval.

2.1 Discrete Time System Stability Using The Invariant Set Principle

Consider a discrete time system of the form:

$$\underline{x}(k+1) = \underline{A}\underline{x}(k) + \underline{B}s \quad (2.1.1)$$

where $\underline{x}(k)$ and \underline{B} are of dimension n , \underline{A} is a constant matrix of dimension $n \times n$, and $s = s(k)$ a scalar. If matrix \underline{A} is convergent (i.e. all eigenvalues within the unit circle) and \underline{B} bounded, then $\underline{x}(k)$ will remain bounded if s is bounded (bounded input-bounded output stability).

Now introduce a related system to that of (2.1.1):

$$s(k+1) = g(s(k)) + d \quad (2.1.2)$$

where $g(s(k))$ is a scalar function of $s(k)$ and d is some bounded scalar. If the s trajectories of (2.1.2) can be shown to lead to an invariant set in finite time, then if this set is bounded, the related system (2.1.1) remains bounded (by the arguments above). An invariant set is defined by [69] as:

A set G is an invariant set for a dynamic system if whenever a point X on a system trajectory is in G , the trajectory remains in G .

If the set G is bounded, then the scalar $s(k)$ is, by definition, bounded also. Because of the convergent nature of matrix \underline{A} , this implies the boundedness of $\underline{x}(k)$. With $s = 0$, of course, the behavior of $\underline{x}(k)$ will be governed solely by the characteristics of matrix \underline{A} .

These concepts will be used in the following controller development to demonstrate the stability properties of the resultant closed loop system.

2.2 Concept of a Discrete Time Sliding Surface

The use of sliding surfaces can be viewed as an alternative method of control when compared with more traditional approaches. These traditional approaches normally seek to place the eigenvalues of the linearized system in such a way that the closed loop system follows the desired behavior. The sliding surface approach instead seeks to constrain the system trajectories in the n dimensional state space to a specified $n - 1$ dimensional hyperplane (the "sliding surface" or "sliding mode") upon which the eigenvalues are specified such that the now constrained motion follows some desired behavior. In the linear system sense, this can be viewed (in a single-input system with full state feedback) as sacrificing one degree of freedom in selecting the eigenvalues and using it to specify one eigenvector.

The motivation behind utilizing the sliding surface approach is that this becomes a possible method for maintaining desired system behavior (normally specified in a linear, time-invariant system context) when disturbances, nonlinearities, time-variant behavior and modeling imperfections are present. Specifically, if a control law can be found which can maintain the system on the sliding surface despite these other factors which act off of the surface, then the system behavior will be governed solely by the dynamics on the surface itself (which are determined by the designer).

For continuous time systems, it is at least conceptually clear that the system state could remain on the sliding surface even in the presence of these imperfections because the control input is "continuously" available to counteract any tendency for the system to move off the surface (motion of the state along the surface is what gives rise to the terms "sliding surface" and "sliding mode"). In a discrete time system, however, the control input is only available at each sampling instant and the system state evolves in an open loop manner between sampling instants. This implies that it will not in general be possible to maintain the system state exactly on the surface, but rather only be kept within a neighborhood of the surface. This behavior in the neighborhood of the surface in the discrete time case will be referred to as a "quasi-sliding mode"

behavior. Figure 2.1 illustrates the difference between sliding and quasi-sliding mode behavior for a two dimensional system.

The inability of the discrete time controller to maintain the system state exactly on the sliding surface requires that if the state is off the surface in some neighborhood around it, this neighborhood constitutes a bounded region. Further, it is necessary to show that when the state is within this bounded region, it implies that the state remains bounded. As will be discussed in Chapter 3, it is apparent that the discrete time quasi-sliding mode behavior may not be achievable for all systems that have controllers based on imperfect models.

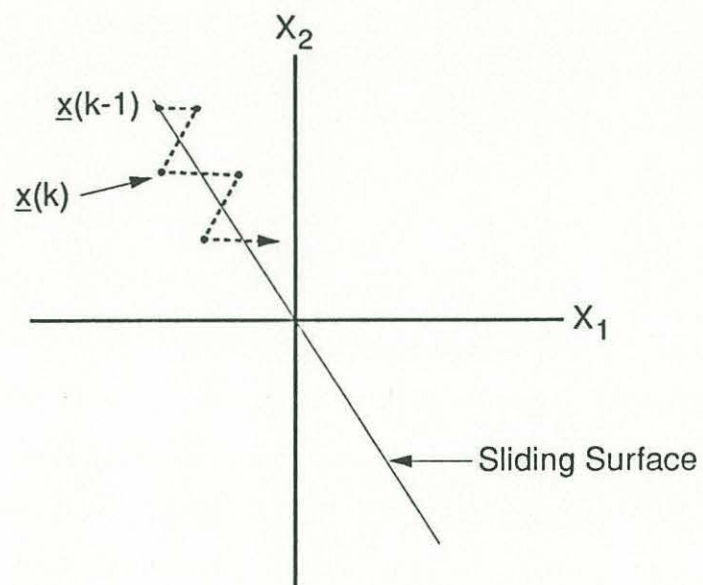
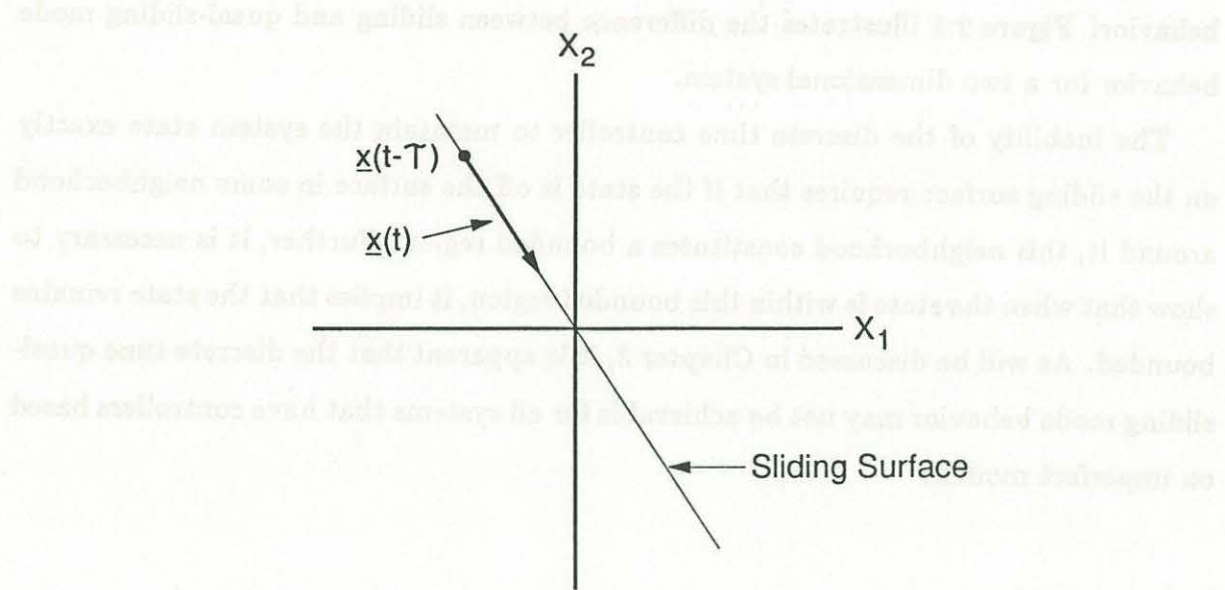


Figure 2.1 Continuous time sliding mode behavior and discrete time quasi-sliding mode behavior for a two dimensional system

2.3 Control Law Formulation

The controller will initially be designed for a single-input discrete time system of the form:

$$\underline{x}(k+1) = \underline{f} + \underline{b}u(k) \quad (2.3.1)$$

where $\underline{x}(k)$, $\underline{f} = \underline{f}(\underline{x}(k), k)$, and $\underline{b} = \underline{b}(\underline{x}(k), k)$ are of dimension n , and $u(k)$ is the scalar control input.

The following assumptions are made:

1. The full state $\underline{x}(k)$ is known at sampling instant k .
2. The system is completely controllable from control input $u(k)$.
3. The functions $\underline{f}(\underline{x}(k), k)$ and $\underline{b}(\underline{x}(k), k)$ are known exactly. This requirement will later be relaxed.

Define the "summarizing scalar" $s = s(k)$:

$$s(k) = \underline{e}^T \underline{x}(k) \quad (2.3.2)$$

where \underline{e} is an n dimensional vector with real valued, constant elements and $s = 0$ defines the sliding surface. The constants that make up \underline{e} are selected such that remaining on the surface automatically implies that certain system properties (e.g. stability) are maintained (the selection of these constants is discussed in the Section 2.4).

Let the control problem be one of the bringing an initial state $\underline{x}(k=0)$ to zero (the regulator problem). Initially let the state lie on the surface (i.e. $s(k=0) = 0$). If $s(k) = 0$, then a control law that causes $s(k+1) = s(k)$ will maintain the system on the surface. Using the system description of (2.3.1) and (2.3.2):

$$s(k+1) = \underline{e}^T \underline{f} + \underline{e}^T \underline{b} u(k) \quad (2.3.3)$$

Setting $s(k+1) = s(k)$ and solving for $u(k)$ in (2.3.3) results in the control law:

$$u(k) = \frac{1}{\underline{e}^T \underline{b}} \{s(k) - \underline{e}^T \underline{f}\} \quad (2.3.4)$$

where it is assumed that \underline{b} does not lie in the null space of \underline{e} (i.e. $\underline{e}^T \underline{b} \neq 0$).

If the initial state does not lie on the surface ($s(k=0) \neq 0$), then the control law must be modified so that the state will reach the surface in finite time and remain on it thereafter. For this purpose, the control law of (2.3.4) is modified to:

$$u(k) = \frac{1}{\underline{e}^T \underline{b}} \{s(k) - \underline{e}^T \underline{f} - \eta \operatorname{sgn}\{s(k)\}\} \quad (2.3.5)$$

where $\operatorname{sgn}\{\cdot\}$ is the "sign" function:

$$\operatorname{sgn}\{x\} = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \end{cases} \quad (2.3.6)$$

and η is a small positive constant. The use of control input (2.3.5) in (2.3.3) results in:

$$s(k+1) = s(k) - \eta \operatorname{sgn}\{s(k)\} \quad (2.3.7)$$

This guarantees that the surface will be reached within $|s(k=0)|/\eta + 1$ sampling periods, but cannot in general maintain the state exactly on the surface $s = 0$. Rather, s will remain within a region bounded by $\pm\eta$ around the surface.

The control law (2.3.4) thus maintains the system state on or within a bounded region of the sliding surface which has been selected because of the desirable characteristics associated with the dynamics on the surface. The next section details the selection of the sliding surface and examines the behavior of the state around the surface $s(k) = 0$. A stability argument for the proposed controller structure can then be made.

2.4.1 Linear Time-Invariant Systems

Consider the system

$$x(k+1) = Ax(k) + Bu(k)$$

(2.4.1)

where $A(k)$ and $B(k)$ are of dimension $n \times n$ and $n \times m$ respectively, $x(k)$ is a vector, $u(k)$ is a scalar, and k is the discrete time index. The system is completely controllable if the rank of the controllability matrix \mathcal{C} is n . The system is completely observable if the rank of the observability matrix \mathcal{O} is n . The system is BIBO stable if the eigenvalues of $A(k)$ are all inside the unit circle.

the closed loop system

$$x(k+1) = \left(1 - \frac{K}{n}\right)x(k) + \frac{K}{n}u(k)$$

(2.4.2)

Along the surface $s = 0$, (2.4.1.2) reduces to

$$x(k+1) = \left(1 - \frac{K}{n}\right)x(k)$$

(2.4.3)

The object is to determine K such that the closed loop system of (2.4.3) has the desired properties on the $n-1$ dimensional hypersurface $s = 0$. This is equivalent to specifying the hypersurface in which $n-1$ eigenvalues lie and the $n-1$ associated eigenvectors that govern the motion on the hypersurface.

2.4 Selection of the Sliding Surface

The selection of constants that comprise the elements of the vector \underline{e} of (2.3.2) which define the sliding surface will now be discussed. The selection of the surface applies to general linear, time-invariant systems and a restricted class of nonlinear, time-variant systems which are both subsets of the general system form of (2.3.1).

2.4.1 Linear, Time-Invariant Systems

Consider the system:

$$\underline{x}(k+1) = \underline{\Phi} \underline{x}(k) + \underline{\Gamma} u(k) \quad (2.4.1.1)$$

where $\underline{x}(k)$ and $\underline{\Gamma}$ are of dimension n , $\underline{\Phi}$ is of dimension $n \times n$, and $u(k)$ is a scalar. As in Section 2.3, the full state $\underline{x}(k)$ is known and the system is completely controllable from input $u(k)$. Using the appropriate form of control law (2.3.4) in (2.4.1.1) yields the closed loop system:

$$\underline{x}(k+1) = \left[\underline{I} - \frac{\underline{\Gamma} \underline{e}^T}{\underline{e}^T \underline{\Gamma}} \right] \underline{\Phi} \underline{x}(k) + \frac{\underline{\Gamma} s(k)}{\underline{e}^T \underline{\Gamma}} \quad (2.4.1.2)$$

Along the surface $s = 0$, (2.4.1.2) reduces to:

$$\underline{x}(k+1) = \left[\underline{I} - \frac{\underline{\Gamma} \underline{e}^T}{\underline{e}^T \underline{\Gamma}} \right] \underline{\Phi} \underline{x}(k) \quad (2.4.1.3)$$

The object is to determine \underline{e} such that the n dimensional system of (2.4.1.3) has the desired properties on the $n - 1$ dimensional hyperplane $s = 0$. This is equivalent to specifying the hyperplane in which $n - 1$ eigenvectors lie and the $n - 1$ associated eigenvalues that govern the motion on the hyperplane.

The first point to notice is that a single constraint between the state variables (being constrained to lie on the surface) combined with the n dimensional system of (2.4.1.3) will result in a singular system because $n + 1$ relations exist for n variables. This means that the system of (2.4.1.3) will have one eigenvalue at zero.

The second point to notice is that, because the system is controllable, a control law exists of the form:

$$u(k) = -\underline{K}^T \underline{x}(k) \quad (2.4.1.4)$$

where \underline{K} is n dimensional and can arbitrarily place the eigenvalues of system (2.4.1.3). Comparison between the control law of (2.3.1.4) (with $s = 0$) and (2.4.1.4) shows that:

$$\underline{K}^T = \frac{\underline{e}^T \underline{\Phi}}{\underline{e}^T \underline{\Gamma}} \quad (2.4.1.5)$$

Now since one eigenvalue must be set at zero, the remaining eigenvalues govern the behavior in the remaining $n - 1$ dimensions (since the additional constraint of remaining on the surface is added to (2.4.1.3)) and can be set as desired using \underline{K} as obtained from any linear eigenvalue placement algorithm [70,71].

Again using (2.4.1.5), it is seen that \underline{e} can be determined to within a multiplicative constant (since $\underline{e}^T \underline{\Gamma}$ is a constant) by using:

$$\underline{e}^T = \alpha \underline{K}^T \underline{\Phi}^{-1} \quad (2.4.1.6)$$

where α is a constant and $\underline{\Phi}$ is assumed invertible. For convenience, α can be selected so that \underline{e} has the form:

$$\underline{e} = [e_1 \cdots e_{n-1} \ 1]^T \quad (2.4.1.7)$$

The selection of \underline{e} is simplified if the system (2.4.1.1) takes on special forms. For example, if (2.4.1.1) is in controller companion form:

$$\underline{x}(k+1) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 \\ a_1 & a_2 & a_3 & \cdots & a_n \end{bmatrix} \underline{x}(k) + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b \end{bmatrix} u(k) \quad (2.4.1.8)$$

then the n dimensional system of (2.4.1.3) reduces to the $n - 1$ dimensional system (with \underline{e} chosen of the form (2.4.1.7)):

$$\underline{x}(k+1) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & -e_1 & -e_2 & \cdots & -e_{n-1} \end{bmatrix} \underline{x}(k) \quad (2.4.1.9)$$

where the last row of the matrix contains the coefficients of the system characteristic equation and thus can easily be adjusted to yield the desired closed loop behavior on the surface (the zero in the first column of the last row corresponding to the zero eigenvalue associated with the constrained motion on the surface).

Assuming that the surface is selected as described above to result in an asymptotically stable system (2.4.1.3), the closed loop system using the control law of (2.3.5) can now be considered:

$$\underline{x}(k+1) = \left[\underline{I} - \frac{\underline{\Gamma} \underline{e}^T}{\underline{e}^T \underline{\Gamma}} \right] \underline{\Phi} \underline{x}(k) + \frac{\underline{\Gamma}}{\underline{e}^T \underline{\Gamma}} \{s(k) - \eta \operatorname{sgn}\{s(k)\}\} \quad (2.4.1.10)$$

where the corresponding s dynamics are given by (2.3.7):

$$s(k+1) = s(k) - \eta \operatorname{sgn}\{s(k)\} \quad (2.4.1.11)$$

As pointed out in Section 2.3, s is guaranteed to reach the surface within $|s(k=0)|/\eta + 1$ sampling periods, and to thereafter remain within a region of $s = 0$ bounded by $\pm\eta$. In the terminology of Section 2.1, the region $s = \pm\eta$ is an invariant set of the system (2.4.1.11). Furthermore, since (2.4.1.3) is an asymptotically stable system, then by analogy with (2.1.1) and the reasoning in Section 2.1, (2.4.1.2) represents a system whose state remains bounded.

In light of the above discussion, the system of (2.4.1.2) can be interpreted as a stable filter in $\underline{x}(k)$ which is driven by input $s(k)$ that represents deviations from the surface $s = 0$.

2.4.2 Nonlinear, Time-Variant Systems

The development of the previous section can be extended to a limited class of nonlinear and time-variant systems. Specifically, the system must be in the form:

$$\begin{aligned} \underline{x}'(k+1) &= \underline{A}' \underline{x}(k) \\ x_n(k+1) &= f + bu(k) \end{aligned} \quad (2.4.2.1)$$

where $\underline{x}'(k) = [x_1(k) \cdots x_{n-1}(k)]^T$, \underline{A}' is a constant $(n-1) \times n$ dimension matrix, and $f = f(\underline{x}(k), k)$ and $b = b(\underline{x}(k), k)$ are scalar functions.

The reason for imposing the additional constraints on the system form can be understood by considering the motivation behind the sliding mode methodology itself. As stressed in Section 2.2, the use of a sliding mode behavior is driven by the desire to remain on a constrained region of the state space where the dynamics follow some desirable properties. These desirable properties are most easily specified in a linear, time-invariant setting because the solutions to these types of systems can be determined explicitly. The nonlinear and time-variant portion of the system must therefore be confined to one dimension only, with the dynamics on the remaining $n-1$ dimensions being confined to those of a linear, time-invariant nature. Further, to allow the control input to effectively "counteract" these nonlinear, time-variant effects, it must act in this dimension only so as to not "corrupt" the remaining $n-1$ dimensions with the resultant nonlinear, time-variant control input. That is, the $\underline{f}(\underline{x}(k), k)$ vector of system (2.3.1) becomes:

$$\underline{f}(\underline{x}(k), k) = \begin{bmatrix} \underline{A}'\underline{x}(k) \\ f(\underline{x}(k), k) \end{bmatrix} \quad (2.4.2.2)$$

and the $\underline{b}(\underline{x}(k), k)$ vector of system (2.3.1) becomes:

$$\underline{b}(\underline{x}(k), k) = [0 \cdots 0 \quad b(\underline{x}(k), k)]^T \quad (2.4.2.3)$$

To determine the vector \underline{e} in the above case (where \underline{e} is of the form (2.4.1.7)), let the control law of (2.3.4.) on the surface be given by:

$$u(k) = \frac{1}{b} \{-f + u'(k)\} \quad (2.4.2.4)$$

where $u'(k)$ is to be determined. The system (2.4.2.1) then becomes:

$$\underline{x}(k+1) = \begin{bmatrix} \underline{A}' \\ \underline{0} \end{bmatrix} \underline{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u'(k) \quad (2.4.2.5)$$

If this modified linear, time-invariant system is controllable from input $u'(k)$, then a control law exists of the form:

$$u'(k) = -\underline{K}^T \underline{x}(k) \quad (2.4.2.6)$$

where \underline{K} is n dimensional and which can arbitrarily place the eigenvalues of system (2.4.2.5) and is determined by standard eigenvalue placement methods as explained in the previous section. As in Section 2.4.1, this implies:

$$\underline{K}^T = \underline{e}^T \begin{bmatrix} \underline{A}' \\ \underline{0} \end{bmatrix} \quad (2.4.2.7)$$

Now defining $\underline{e}' = [e_1 \cdots e_{n-1}]^T$, this is equivalent to:

$$\underline{K}^T = \underline{e}'^T \underline{A}' \quad (2.4.2.8)$$

If the pseudoinverse of \underline{A} exists (i.e. if $\underline{A} \underline{A}^T$ is nonsingular), then:

$$\underline{e}'^T = \underline{K}^T \underline{A}'^T (\underline{A}' \underline{A}'^T)^{-1} \quad (2.4.2.9)$$

and thus:

$$\underline{e} = [K^T \underline{A}'^T (\underline{A}' \underline{A}'^T)^{-1} \ 1]^T \quad (2.4.2.10)$$

By comparison with Section 2.4.1, this can be viewed as a more general case of the system (2.4.2.1) being in controller companion form. That is, if:

$$\underline{A}' = [0 \ \underline{I}] \quad (2.4.2.11)$$

then (2.4.2.1) will be in controller companion form which permits a more simple determination of \underline{e} as shown in the previous section.

The final control law thus takes the form:

$$u(k) = \frac{1}{b} \{s(k) - f - \underline{e}'^T \underline{A}' \underline{x}(k)\} \quad (2.4.2.12)$$

To account for an initial state off the surface, the control law is augmented with the discontinuous term $\eta \operatorname{sgn}\{s(k)\}$ as:

$$u(k) = \frac{1}{b} \{s(k) - f - \underline{e}'^T \underline{A}' \underline{x}(k) - \eta \operatorname{sgn}\{s(k)\}\} \quad (2.4.2.13)$$

which maintains s within a bounded region of thickness $\pm\eta$ around the surface. This results in the closed loop system:

$$\underline{x}(k+1) = \begin{bmatrix} \underline{A}' \\ -\underline{e}'^T \underline{A}' \end{bmatrix} \underline{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \{s(k) - \eta \operatorname{sgn}\{s(k)\}\} \quad (2.4.2.14)$$

with the corresponding s dynamics:

$$s(k+1) = s(k) - \eta \operatorname{sgn}\{s(k)\} \quad (2.4.2.15)$$

System stability follows exactly as in Section 2.4.1.

EXAMPLE 2.1

The concepts covered thus far will be covered in this example to illustrate the application of the method. The particular constants selected will correspond to those to be used in the experimental study in the latter part of this thesis. Consider the second order system

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -7.5 & -0.5 \end{bmatrix} x(t) + \begin{bmatrix} 0.22 \\ 0.02 \end{bmatrix} u(t) = \begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix}$$

(2.1.1)

Using the notation of Section 2.4.1

$$A = \begin{bmatrix} 0 & 1 \\ -7.5 & -0.5 \end{bmatrix}$$

(2.1.2)

$$B = \begin{bmatrix} 0.22 \\ 0.02 \end{bmatrix}$$

(2.1.3)

$$C = \begin{bmatrix} 0.002 & 0 \end{bmatrix}$$

(2.1.4)

Let the vector z have the form

$$z = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

(2.1.5)

and the normalized vector $\hat{z}(t)$ is therefore

EXAMPLE 2.1

The concepts covered thus far will be covered in this example to illustrate the application of the method. The particular constants selected will correspond to those to be used in the experimental study in the latter part of this thesis. Consider the second order system:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.22 \\ 0 & 0.95 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{0.0066}{1+0.05|x_2(k)|} \end{bmatrix} u(k) \quad (\text{Ex. 2.1.1})$$

Using the notation of Section 2.4.2:

$$\underline{A}' = \begin{bmatrix} 1 & 0.22 \end{bmatrix} \quad (\text{Ex. 2.1.2})$$

$$f = 0.95x_2(k) \quad (\text{Ex. 2.1.3})$$

$$b = \frac{0.0066}{1 + 0.05 |x_2(k)|} \quad (\text{Ex. 2.1.4})$$

Let the vector \underline{e} have the form:

$$\underline{e} = \begin{bmatrix} \lambda & 1 \end{bmatrix}^T \quad (\text{Ex. 2.1.5})$$

and the summarizing scalar $s(k)$ is therefore:

$$s(k) = \underline{e}^T \underline{x}(k) = \lambda x_1(k) + x_2(k)$$

(Ex. 2.1.6)

Initially let the control law be given by (2.4.2.12):

$$u(k) = \frac{1 + 0.05 |x_2(k)|}{0.0066} \{s(k) - \lambda x_1(k) - (0.22\lambda + 0.95)x_2(k)\}$$

(Ex. 2.1.7)

Substitution of control law (Ex. 2.1.7) into system (Ex. 2.1.1) on the surface ($s(k) = 0$) results in the modified linear, time-invariant system:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.22 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u'(k)$$

(Ex. 2.1.8)

where:

$$u'(k) = -\lambda \{x_1(k) + 0.22x_2(k)\}$$

(Ex. 2.1.9)

Note that the system of (Ex. 2.1.8) is controllable, with controllability matrix:

$$\underline{C} = \begin{bmatrix} 0 & 0.22 \\ 1 & 0 \end{bmatrix}$$

(Ex. 2.1.10)

As shown in Section 2.4.2, the input $u'(k)$ of (Ex. 2.1.9) can be written as:

$$u'(k) = -\underline{K}^T \underline{x}(k)$$

(Ex. 2.1.11)

where:

$$\underline{K}^T = \underline{e}^T \begin{bmatrix} \underline{A}' \\ 0 \end{bmatrix}$$

(Ex. 2.1.12)

The next step is to specify the desired eigenvalues of the closed loop system. As discussed earlier, one eigenvalue must be set to zero to reflect the constraint of remaining on the surface. The remaining eigenvalue dictates the behavior on the surface itself, and the value 0.89 is selected so that the dynamics on the surface are asymptotically stable and are of a low-pass filter nature. Using an eigenvalue placement technique to select \underline{K} so that the eigenvalues (0.0, 0.89) are achieved results in:

$$\underline{K} = [0.5 \quad 0.11]^T$$

(Ex. 2.1.13)

To determine λ , note from (Ex. 2.1.5) and (2.4.2.10) that:

$$\lambda = \underline{K}^T \underline{A}'^T (\underline{A}' \underline{A}'^T)^{-1}$$

(Ex. 2.1.14)

which gives a value of $\lambda = 0.5$.

The control law of (Ex. 2.1.7) can now be expressed as:

$$u(k) = \frac{1 + 0.05 |x_2(k)|}{0.0066} \{s(k) - 0.5x_1(k) - 1.06x_2(k)\}$$

(Ex. 2.1.15)

To account for an initial state that is off the surface, the final control law is augmented with a discontinuous term as in (2.4.2.13):

$$u(k) = \frac{1 + 0.05 |x_2(k)|}{0.0066} \{s(k) - 0.5x_1(k) - 1.06x_2(k) - \eta \operatorname{sgn}\{s(k)\}\} \quad (\text{Ex. 2.1.16})$$

where η is a small positive constant.

A simulation was performed with initial condition $\underline{x}(k=0) = [-10 \ 5.5]^T$ using $\eta = 0.1$. This results in $s(k=0) = 0.5$ and thus the state is initially off the surface. Figure 2.2 illustrates the evolution of the state trajectory in the phase plane. The trajectory is composed of two modes of behavior – first, the movement of the state towards the surface and second, the movement of the state on or close to the surface in a quasi-sliding mode manner toward the origin. The first mode is largely determined by the value for η , and the second mode by the eigenvalue chosen to govern behavior on the surface.

Figure 2.3 shows the behavior of s with time. From the initial value $s(k=0) = 0.5$ with $\eta = 0.1$, the s trajectory reaches the surface within 6 sampling periods as predicted, and thereafter remains within a distance $\pm\eta$ from the surface.

The implications of $s(k)$ remaining with this bounded region (rather than equal to zero) on the final values of $x_1(k)$ and $x_2(k)$ are illustrated graphically in Figure 2.4. Note from (Ex. 2.1.1) and the control law of (Ex. 2.1.16) that the closed loop system is:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.22 \\ -\lambda & -0.22\lambda \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ s(k) - \eta \operatorname{sgn}\{s(k)\} \end{bmatrix} \quad (\text{Ex. 2.1.17})$$

Now for the general linear, time-invariant discrete time system:

$$\underline{x}(k+1) = \underline{\Phi} \underline{x}(k) + \underline{\Gamma} u \quad (\text{Ex. 2.1.18})$$

where u is a bounded driving term. The solution takes the form:

$$\underline{x}(k) = \underline{\Phi}^k \underline{x}(k=0) + \sum_{i=0}^{k-1} \{ \underline{\Phi}^i \underline{\Gamma} u \} \quad (\text{Ex. 2.1.19})$$

For a convergent matrix $\underline{\Phi}$, the steady state solution becomes:

$$\underline{x}(k = \infty) = \sum_{i=0}^{\infty} \{ \underline{\Phi}^i \underline{\Gamma} u \} \quad (\text{Ex. 2.1.20})$$

The following bound then applies:

$$\|\underline{x}(k = \infty)\| \leq \{ \|\underline{\Gamma}\| + \|\underline{\Phi}\| + \|\underline{\Phi}\|^2 + \|\underline{\Phi}\|^3 + \dots \} \|\underline{\Gamma} u\| \quad (\text{Ex. 2.1.21})$$

which implies:

$$\|\underline{x}(k = \infty)\| \leq \{ 1 + p_{maz} + p_{maz}^2 + p_{maz}^3 + \dots \} \|\underline{\Gamma} u\| \quad (\text{Ex. 2.1.22})$$

where p_{maz} is the largest absolute magnitude eigenvalue of $\underline{\Phi}$. Since $\underline{\Phi}$ is a convergent matrix, $p_{maz} < 1$ and so (Ex. 2.1.22) becomes:

$$\|\underline{x}(k = \infty)\| \leq \frac{1}{1 - |p_{maz}|} \|\underline{\Gamma} u\| \quad (\text{Ex. 2.1.23})$$

where a series expansion definition has been used.

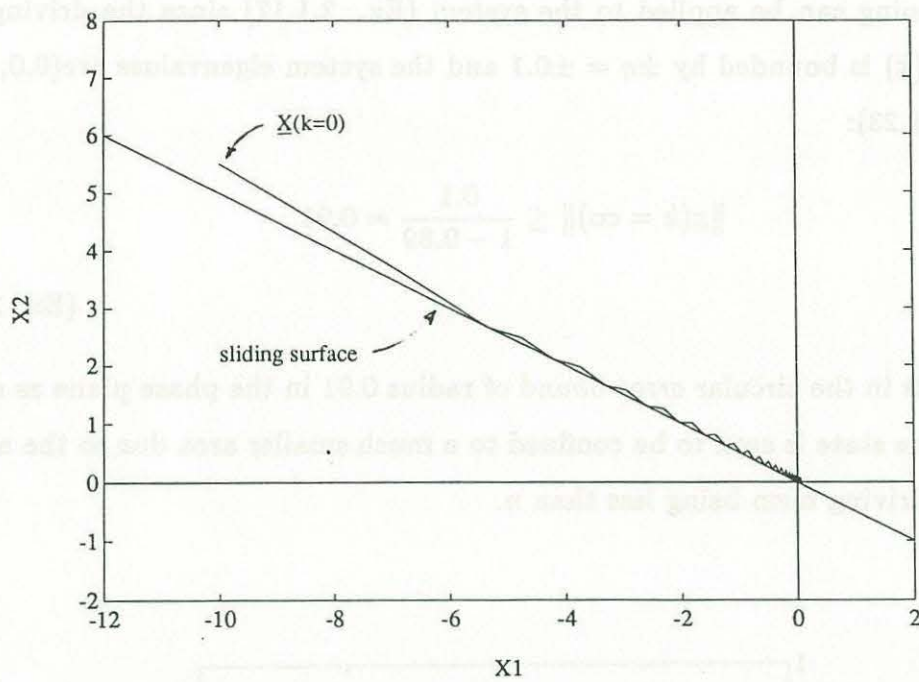


Figure 2.2 Phase plane trajectory

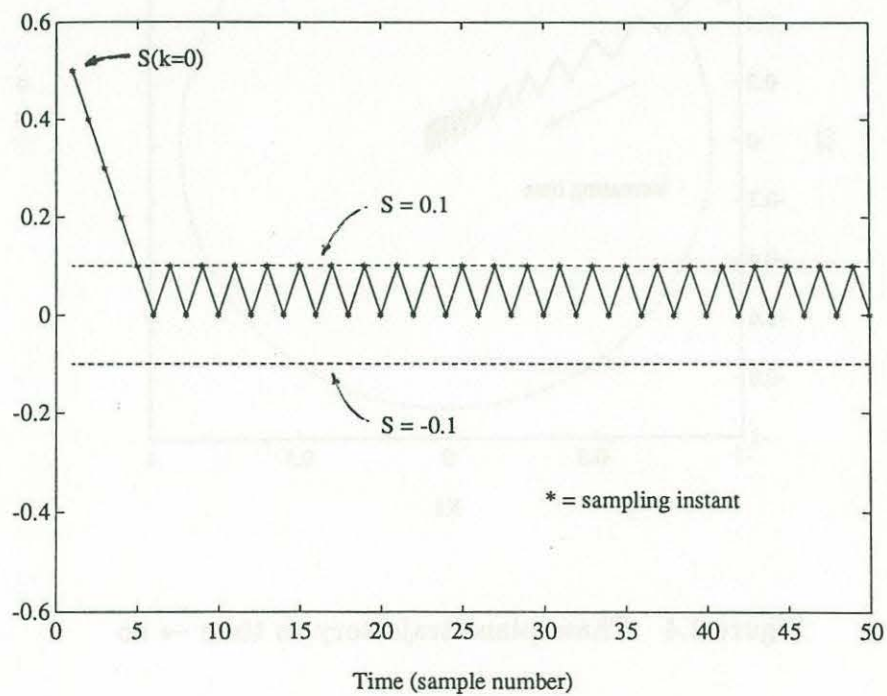


Figure 2.3 S trajectory

This reasoning can be applied to the system (Ex. 2.1.17) since the driving term $s(k) = \eta \operatorname{sgn}(k)$ is bounded by $\pm\eta = \pm 0.1$ and the system eigenvalues are $(0.0, 0.89)$. Using (Ex. 2.1.23):

$$\|\underline{x}(k = \infty)\| \leq \frac{0.1}{1 - 0.89} = 0.91$$

(Ex. 2.1.24)

This results in the circular error bound of radius 0.91 in the phase plane as seen in Figure 2.4. The state is seen to be confined to a much smaller area due to the average value for the driving term being less than η .

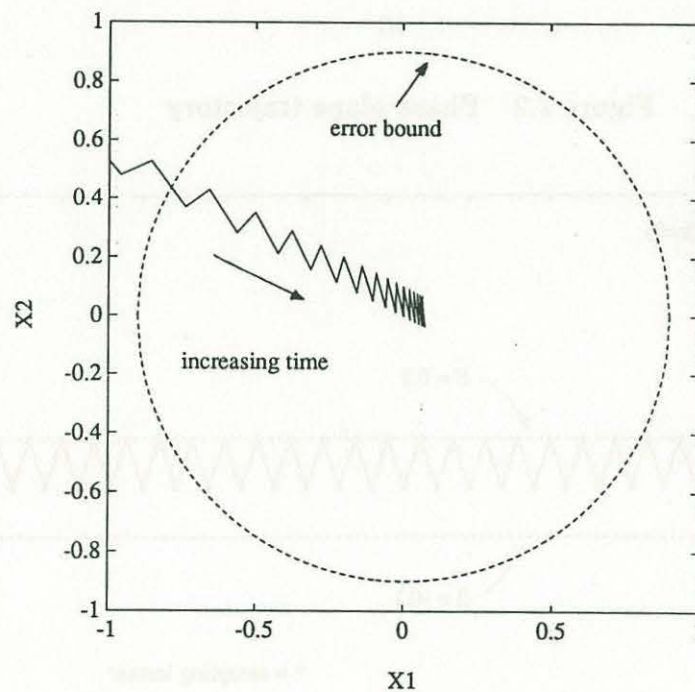


Figure 2.4 Phase plane trajectory as time $\rightarrow \infty$

2.5 Extension To Trajectory Following Controller Form

The control law formulation of Section 2.3 was designed for the regulator problem which seeks to bring an initial state $\underline{x}(k=0)$ to zero. This control law can also be extended to apply to the more general trajectory following or "tracking" problem. In the trajectory following problem, the goal is to have the state $\underline{x}(k)$ follow a desired state $\underline{x}_d(k)$ as closely as possible.

For this purpose the summarizing scalar of Section 2.3 is redefined as:

$$s(k) = \underline{e}^T \tilde{\underline{x}}(k) \quad (2.5.1)$$

where $\tilde{\underline{x}}(k) = \underline{x}(k) - \underline{x}_d(k)$. Following the same procedure as in Section 2.3 and using the nonlinear, time-variant system of Section 2.4.2, the control law becomes:

$$u(k) = \frac{1}{b} \left\{ s(k) - f - \underline{e}^T \underline{A}' \underline{x}(k) + \underline{e}^T \underline{x}_d(k+1) \right\} \quad (2.5.2)$$

This yields the closed loop system on the surface:

$$\underline{x}(k+1) = \begin{bmatrix} \underline{A} \\ -\underline{e}'^T \underline{A}' \end{bmatrix} \underline{x}(k) + \begin{bmatrix} \underline{0} \\ \underline{e}^T \end{bmatrix} \underline{x}_d(k+1) \quad (2.5.3)$$

With $\underline{x}_d(k+1)$ being selected as a bounded vector for all k , the stability arguments of Section 2.4 remain the same. As in (2.4.2.13), the control law of (2.5.2) is augmented with the term $\eta \operatorname{sgn}\{s(k)\}$ to account for the general case when the state is not on the surface at $k=0$. To make the controller development as general as possible, the summarizing scalar $s(k)$ will hereafter be defined by (2.5.1).

Regarding the determination of the n dimension vector $\underline{x}_d(k)$, a single-input discrete time system can in general only satisfy one component of this vector from one sampling instant to the next. That is, to make $\underline{x}(k) = \underline{x}_d(k)$, a control law must exist that can match each component of the state vector at each sampling instant which implies some "consistency" requirement must exist between the components of $\underline{x}_d(k)$. The exact form of this consistency requirement depends on the structure of the system being analyzed. For example, for the system given by (2.4.2.1), if $x_{n_d}(k)$ is the one element of $\underline{x}_d(k)$ specified, then the remaining elements of $\underline{x}_d(k)$ are given by:

$$\underline{x}'_d(k) = \underline{A}' \underline{x}_d(k-1) \quad (2.5.4)$$

where $\underline{x}'_d(k) = [x_{1_d}(k) \cdots x_{n-1_d}(k)]^T$. When this consistency requirement is met, (2.5.3) becomes:

$$\tilde{\underline{x}}(k+1) = \begin{bmatrix} \underline{A}' \\ -\underline{e}'^T \underline{A}' \end{bmatrix} \tilde{\underline{x}}(k) \quad (2.5.5)$$

and thus reflects the error dynamics between the actual state $\underline{x}(k)$ and desired state $\underline{x}_d(k)$. A similar development can be carried out using the linear, time-invariant system form of Section 2.4.1.

Finally, there is an implicit assumption that $x_d(k+1)$ is known at sampling instant k , which is reasonable since the effect of $u(k)$ cannot be observed until sampling instant $k+1$.

EXAMPLE 2.2

The controller of Example 2.1 can be modified to the more general trajectory following form by using the summarizing scalar definition of (2.5.1) and changing the control law to:

$$u(k) = \frac{1 + 0.05 |x_2(k)|}{0.0066} \{s(k) - 0.5x_1(k) - 1.06x_2(k) + 0.5x_{1d}(k+1) + x_{2d}(k+1) - \eta \operatorname{sgn}\{s(k)\}\} \quad (\text{Ex. 2.2.1})$$

Let the desired trajectory for $x_{2d}(k)$ be:

$$x_{2d}(k) = 10 \sin(0.066k) \quad (\text{Ex. 2.2.2})$$

Recall from Example 2.1 that:

$$\underline{A}' = [1 \quad 0.22] \quad (\text{Ex. 2.2.3})$$

and thus the trajectory consistency requirement of (2.5.4) is of the form:

$$x_{1d}(k) = x_{1d}(k-1) + 0.22x_{2d}(k-1) \quad (\text{Ex. 2.2.4})$$

As explained in Section 2.5, the system behavior should be identical to the regulator problem of Example 2.1 except that the error state $\tilde{x}(k)$ replaces the system state $\underline{x}(k)$. Figure 2.5 shows the error phase plane which is identical to the phase plane of Figure 2.2. The error behavior follows the two modes discussed in Example 2.1 - first, reaching the surface and second, remaining on or near the surface and converging to a region near the origin.

Figure 2.6 and 2.7 respectively illustrate the trajectory following performance of both states. The bound on the errors can again be obtained exactly as done in Example 2.1.

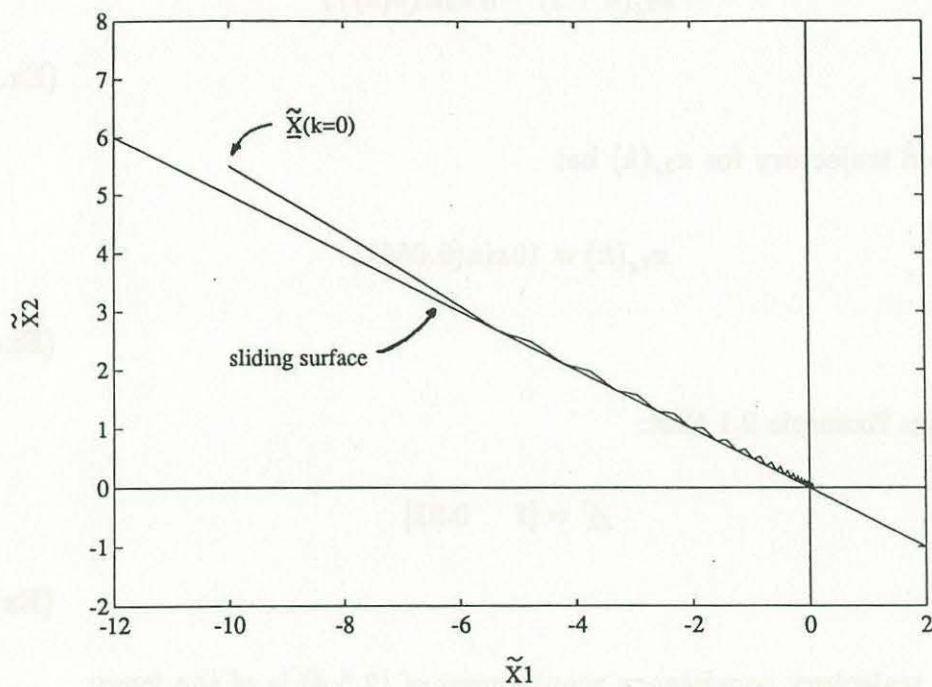


Figure 2.5 Error phase plane trajectory

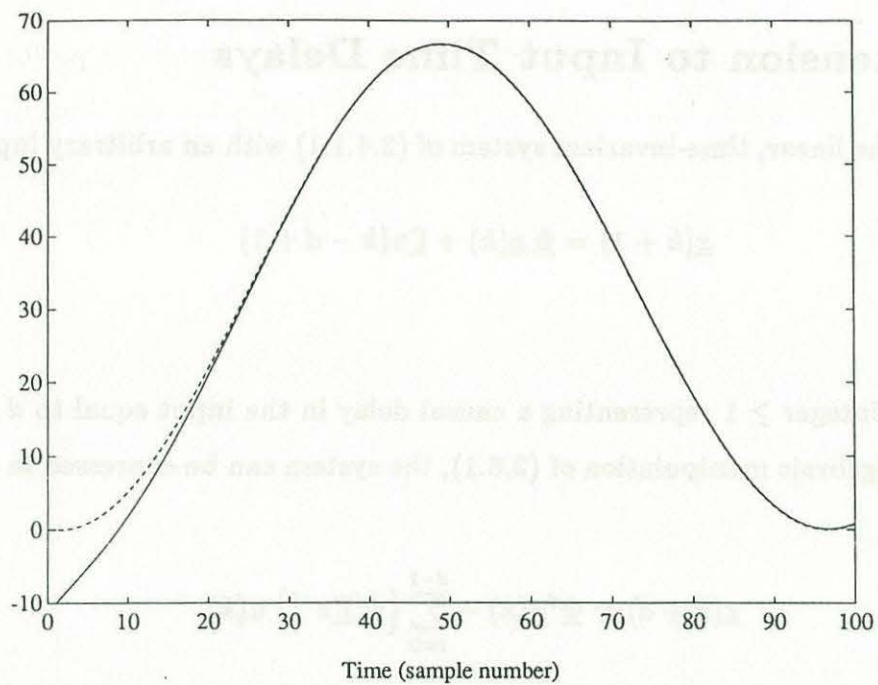


Figure 2.6 Actual (-) and desired (--) x_1

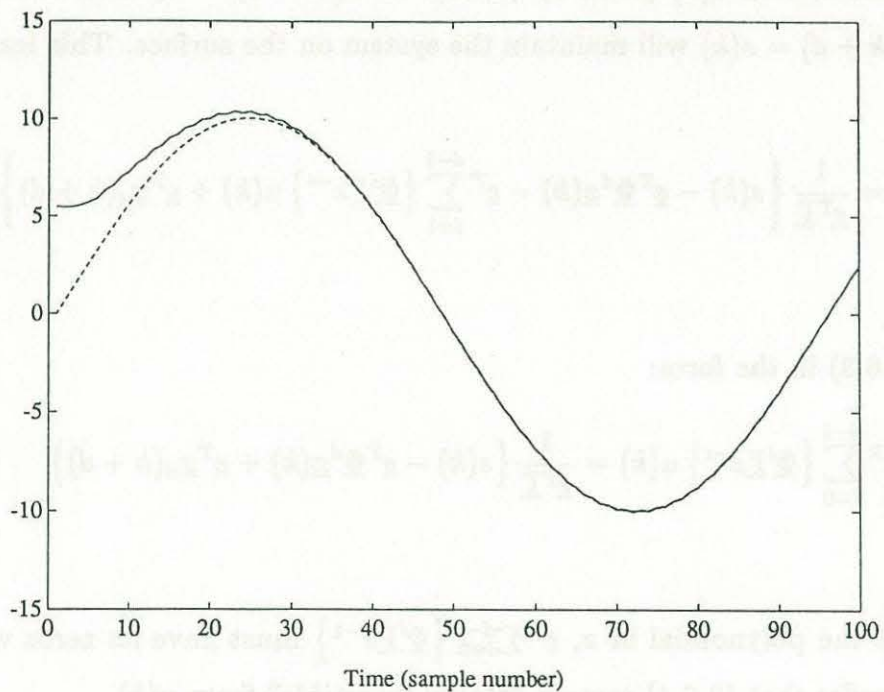


Figure 2.7 Actual (-) and desired (--) x_2

2.6 Extension to Input Time Delays

Consider the linear, time-invariant system of (2.4.1.1) with an arbitrary input delay:

$$\underline{x}(k+1) = \underline{\Phi} \underline{x}(k) + \underline{\Gamma} u(k-d+1) \quad (2.6.1)$$

where d is an integer ≥ 1 representing a causal delay in the input equal to d sampling periods. By algebraic manipulation of (2.6.1), the system can be expressed in predictor form:

$$\underline{x}(k+d) = \underline{\Phi}^d \underline{x}(k) + \sum_{i=0}^{d-1} \{ \underline{\Phi}^i \underline{\Gamma} z^{-i} \} u(k) \quad (2.6.2)$$

where z^{-1} is the unit delay operator. As in Section 2.3, assume initially that the state lies on the surface with $s(k)$ given by (2.5.1). If $s(k=0) = 0$, then a control law that causes $s(k+d) = s(k)$ will maintain the system on the surface. This leads to the control law:

$$u(k) = \frac{1}{\underline{e}^T \underline{\Gamma}} \left\{ s(k) - \underline{e}^T \underline{\Phi}^d \underline{x}(k) - \underline{e}^T \sum_{i=1}^{d-1} \{ \underline{\Phi}^i \underline{\Gamma} z^{-i} \} u(k) + \underline{e}^T \underline{x}_d(k+d) \right\} \quad (2.6.3)$$

By writing (2.6.3) in the form:

$$\underline{e}^T \sum_{i=0}^{d-1} \{ \underline{\Phi}^i \underline{\Gamma} z^{-i} \} u(k) = \frac{1}{\underline{e}^T \underline{\Gamma}} \left\{ s(k) - \underline{e}^T \underline{\Phi}^d \underline{x}(k) + \underline{e}^T \underline{x}_d(k+d) \right\} \quad (2.6.4)$$

it is clear that the polynomial in z , $\underline{e}^T \sum_{i=0}^{d-1} \{ \underline{\Phi}^i \underline{\Gamma} z^{-i} \}$ must have its zeros within the unit circle in order that (2.6.4) remain "stably invertible" from $u(k)$.

Substitution of control law (2.6.3) into system (2.6.1) leads to:

$$\underline{x}(k+d) = \left[\underline{I} - \frac{\underline{\Gamma} \underline{e}^T}{\underline{e}^T \underline{\Gamma}} \right] \underline{\Phi} \underline{x}(k+d-1) + \frac{\underline{\Gamma} \underline{e}^T \underline{x}_d(k+d)}{\underline{e}^T \underline{\Gamma}} \quad (2.6.5)$$

As discussed in Section 2.5, for consistent desired trajectories this can be expressed as:

$$\tilde{\underline{x}}(k+d) = \left[\underline{I} - \frac{\underline{\Gamma} \underline{e}^T}{\underline{e}^T \underline{\Gamma}} \right] \underline{\Phi} \tilde{\underline{x}}(k+d-1) \quad (2.6.6)$$

As in (2.3.5), the control law is again augmented with the term $\eta \operatorname{sgn}\{s(k)\}$ to account for the general case when the state is not on the surface at $k=0$.

The same approach can be applied in principle to the nonlinear, time-variant system of Section 2.4.2. The derivation of a control law of the form (2.6.3) will be, in general, more difficult (if even possible) because of the nonlinearities and/or time-variant behavior present. It will also be more difficult to show that the corresponding nonlinear and/or time-variant expression to (2.6.4) is "stably invertible" from $u(k)$.

Example 2.3

Let the system of Example 2.1 be again considered but with an additional delay in the input equal to one sampling period:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.22 \\ 0 & 0.95 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{0.0066}{1+0.05|x_2(k)|} \end{bmatrix} u(k-1) \quad (\text{Ex. 2.3.1})$$

The system can be expressed in predictor form:

$$\begin{bmatrix} x_1(k+2) \\ x_2(k+2) \end{bmatrix} = \begin{bmatrix} 1 & 0.429 \\ 0 & 0.9025 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \underline{\Gamma}'' u(k-1) + \underline{\Gamma}' u(k) \quad (\text{Ex. 2.3.2})$$

where:

$$\underline{\Gamma}'' = \begin{bmatrix} 0.22 \\ 0.95 \end{bmatrix} \left\{ \frac{0.0066}{1+0.05|x_2(k)|} \right\} \quad (\text{Ex. 2.3.3})$$

and:

$$\underline{\Gamma}' = \begin{bmatrix} 0 \\ \frac{0.0066}{1+0.05|0.95x_2(k)+\frac{0.0066u(k-1)}{1+0.05|x_2(k)|}} \end{bmatrix} \quad (\text{Ex. 2.3.4})$$

The control law becomes:

$$u(k) = \frac{1}{\underline{e}^T \underline{\Gamma}'} \left\{ s(k) - \underline{e}^T \begin{bmatrix} 1 & 0.429 \\ 0 & 0.9025 \end{bmatrix} \underline{x}(k) - \underline{e}^T \underline{\Gamma}'' u(k-1) + \underline{e}^T \underline{x}_d(k+2) \right\} \quad (\text{Ex. 2.3.5})$$

where $\underline{e} = [0.5 \ 1]^T$ and the desired trajectory consistency requirement is:

$$\underline{x}_{1_d}(k+2) = \underline{x}_{1_d}(k+1) + 0.22x_{2_d}(k+1)$$

(Ex. 2.3.6)

As stated in Section 2.6, the nonlinearities present in $\underline{\Gamma}''$ and $\underline{\Gamma}'$ do not allow a straightforward determination of the stable invertible nature of the resulting control law. In this case, some additional insight can be obtained by considering a related linear system. For the linearized version (around $x_2(k) = 0$) of (Ex. 2.3.1) where $\underline{\Gamma} = [0 \ 0.0066]^T$, the polynomial of interest to make the stable invertible determination is:

$$\underline{e}^T \{ \underline{\Gamma} + \Phi \underline{\Gamma} z \} = 0$$

(Ex. 2.3.7)

or:

$$1 + 1.06z = 0$$

(Ex. 2.3.8)

which has a zero $z = -0.9434$ that, being inside the unit circle, implies the stable invertible nature of $u(k)$. This linearized analysis is, of course, only valid in the vicinity of $x_2(k) = 0$ and cannot be generalized to the entire state space.

Chapter 3

Discrete Time Control of Uncertain Dynamic Systems Using Sliding Surfaces

The development of the preceding chapter can be extended to deal with a class of uncertain dynamic systems using the sliding surface approach. The uncertainties considered here are external disturbances and errors in the system model. As discussed in Sections 2.2 and 2.4.2, the sliding surface approach is based on the premise that all nonlinear and time-variant effects as well as disturbances reside in a single dimension of the n dimensional state space, with the remaining dynamics on the $n - 1$ dimensions being known. Uncertainties can also be handled by similarly confining them to a single dimension. As in Section 2.4.2 which dealt with nonlinear, time-variant systems, the system must be in the form:

$$\begin{aligned}\underline{x}'(k+1) &= \underline{A}'\underline{x}(k) \\ x_n(k+1) &= f + bu(k) + d(k)\end{aligned}\tag{3.1}$$

where $\underline{x}'(k) = [x_1(k) \cdots x_{n-1}(k)]^T$, \underline{A}' is a constant $(n-1) \times n$ dimension matrix, $f = f(\underline{x}(k), k)$ and $b = b(\underline{x}(k), k)$ are scalar functions, and $d(k)$ is a scalar disturbance. Note that this encompasses the most general case where $f(\underline{x}(k), k)$ and $b(\underline{x}(k), k)$ can be nonlinear and time-variant.

By limiting the system description to that of (3.1), the method for selection of the sliding surface is identical to that given in Section 2.4.2. The chapter begins by considering disturbances $d(k)$, uncertainties in system dynamics $f(\underline{x}(k), k)$, and uncertainties in control gain $b(\underline{x}(k), k)$. The trajectory following controller form will be used for the most general applicability to both regulator and trajectory following problems. Although in principle the arguments presented can be extended into the arbitrarily delayed input case, for simplicity of notation only the single input delay system of (3.1) will explicitly be covered.

It should be noted at the outset of this chapter that the development of discrete time controllers that can stabilize uncertain discrete time systems is quite difficult and still the subject of much current research. The difficulty principally lies in the inability to affect the system between sampling instants - that is, the system is essentially operating in an open loop manner between sampling instants. In the continuous time development of sliding mode control, for example, it is possible to design a control law which can reach and maintain the system state on the sliding surface despite the presence of errors in modeling or disturbances. This means that the state can also be driven to a hyperplane (or a bounded region around one) with known stability properties. The inability to rigorously design such a control law in discrete time to accomplish this task makes the development fundamentally different although similar in appearance.

3.1 Disturbances

Let the disturbance $d(k)$ of system (3.1) be unknown but bounded:

$$D \geq |d(k)| \quad (3.1.1)$$

where D is a bounded scalar. The appropriate version of the trajectory following control law of (2.5.2) is augmented with a new discontinuous term $K \operatorname{sgn}\{s(k)\}$:

$$u(k) = \frac{1}{b} \left\{ s(k) - f - \underline{e}^T \underline{A}' \underline{x}(k) + \underline{e}^T \underline{x}_d(k+1) - K \operatorname{sgn}\{s(k)\} \right\} \quad (3.1.2)$$

where K is to be determined.

The s dynamics which result from the closed loop system of (3.1) and (3.1.2) are:

$$s(k+1) = s(k) - K \operatorname{sgn}\{s(k)\} + d(k) \quad (3.1.3)$$

If K is chosen as:

$$K = D + \eta \quad (3.1.4)$$

(where η is the small positive constant introduced previously) then the surface will be reached in at most $|s(k=0)|/\eta + 1$ sampling periods and will thereafter remain within a distance $\pm(2D + \eta)$ from the surface.

The use of the discontinuous term $K \operatorname{sgn}\{s(k)\}$ introduces a high frequency (Nyquist rate) content into the control signal which may not be achievable by the controller actuator or which could excite unmodeled dynamics in the system being controlled. The

control input can therefore be smoothed by introducing a “boundary layer” or region around the sliding surface of thickness $\Phi > 0$ which is motivated by the continuous time development of [5]. Figure 3.1 illustrates the boundary layer concept for a two dimensional system.

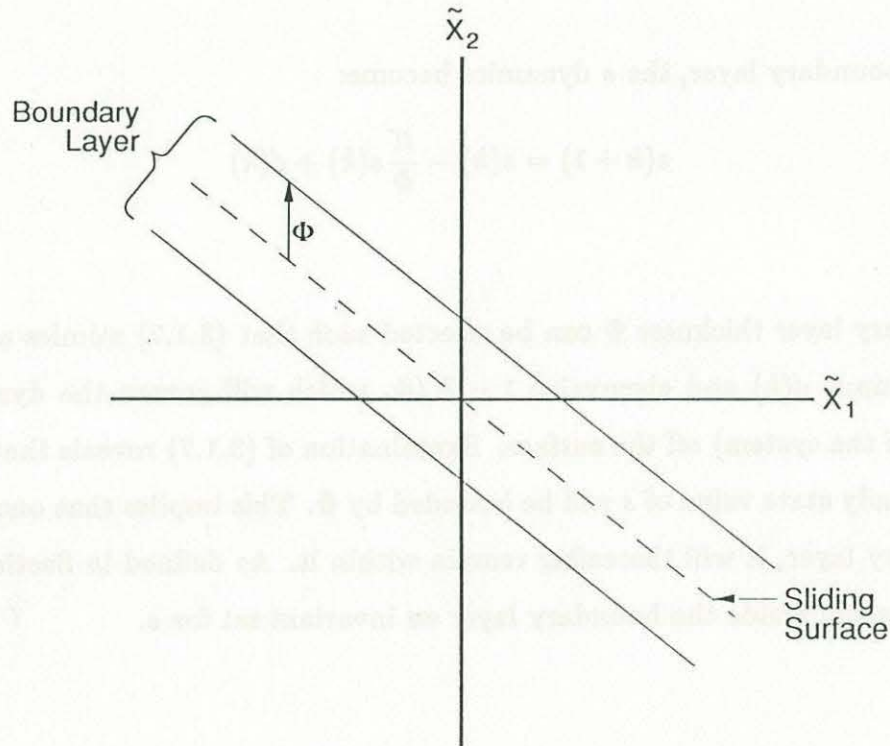


Figure 3.1 Sliding surface with boundary layer for a two dimensional system

When the system state is “inside” the boundary layer ($|s| < \Phi$), then the control law of (3.1.2) is changed to:

$$u(k) = \frac{1}{b} \left\{ s(k) - f - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1) + \right. \\ \left. -Ksat \left\{ \frac{s(k)}{\Phi} \right\} \right\} \quad (3.1.5)$$

where $\text{sat}\{\cdot\}$ is the "saturation" function:

$$\text{sat}\{x\} = \begin{cases} 1 & \text{if } x > 1 \\ -1 & \text{if } x < -1 \\ x & \text{otherwise} \end{cases} \quad (3.1.6)$$

Inside the boundary layer, the s dynamics become:

$$s(k+1) = s(k) - \frac{K}{\Phi} s(k) + d(k) \quad (3.1.7)$$

The boundary layer thickness Φ can be selected such that (3.1.7) mimics a first order filter with input $d(k)$ and eigenvalue $1 - K/\Phi$, which will govern the dynamics of s (and thus of the system) off the surface. Examination of (3.1.7) reveals that if $\Phi \geq K$, then the steady state value of s will be bounded by Φ . This implies that once s reaches the boundary layer, it will thereafter remain within it. As defined in Section 2.1, this makes the region inside the boundary layer an invariant set for s .

Example 3.1

Consider again the system (Ex. 2.1.1) of Example 2.1 but with a disturbance $d(k)$:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.22 \\ 0 & 0.95 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{0.0066}{1+0.05|x_2(k)|} \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d(k) \quad (\text{Ex. 3.1.1})$$

where $|d(k)| \leq 0.2$. selecting $D = 0.2$ and $n = 0.1$ results in $K = 0.3$ from (3.4.1). Further, since the dynamics on the surface are characterized by \underline{e} and assigned an eigenvalue of 0.89, let the dynamics within the boundary layer also be characterized by this eigenvalue. Thus, the eigenvalue controlling the s dynamics is assigned as $p = 0.89$:

$$1 - \frac{0.3}{\Phi} = 0.89 \quad (\text{Ex. 3.1.2})$$

or $\Phi \cong 2.73$.

The simulation of the trajectory following controller of Example 2.2 was repeated but with disturbance:

$$d(k) = 0.2 \sin(\pi k/2) \quad (\text{Ex. 3.1.3})$$

for which the only knowledge available to the controller was the absolute bound D . The initial condition was changed to $\underline{x}(k=0) = [-10 \ 10]^T$ so that $s(k=0)$ was outside the boundary layer in order to illustrate the effect of changing from the discontinuous control law (3.1.2) outside the boundary layer to the smoothed control law (3.1.5) inside the boundary layer.

The error phase plane is shown in Figure 3.2. As before, the behavior of s is characterized by two modes - first, attraction to the boundary layer, and next, remaining within the boundary layer and moving towards a bounded region around the origin. Notice that as the eigenvalue of the s dynamics approaches one (i.e. as the bandwidth is lowered), the magnitude of Φ increases, hence making the boundary layer larger as well. The invariant set region thus increases as the bandwidth is lowered which implies a reduction in tracking performance. The smoothed control law is therefore achieved at the cost of reduced performance.

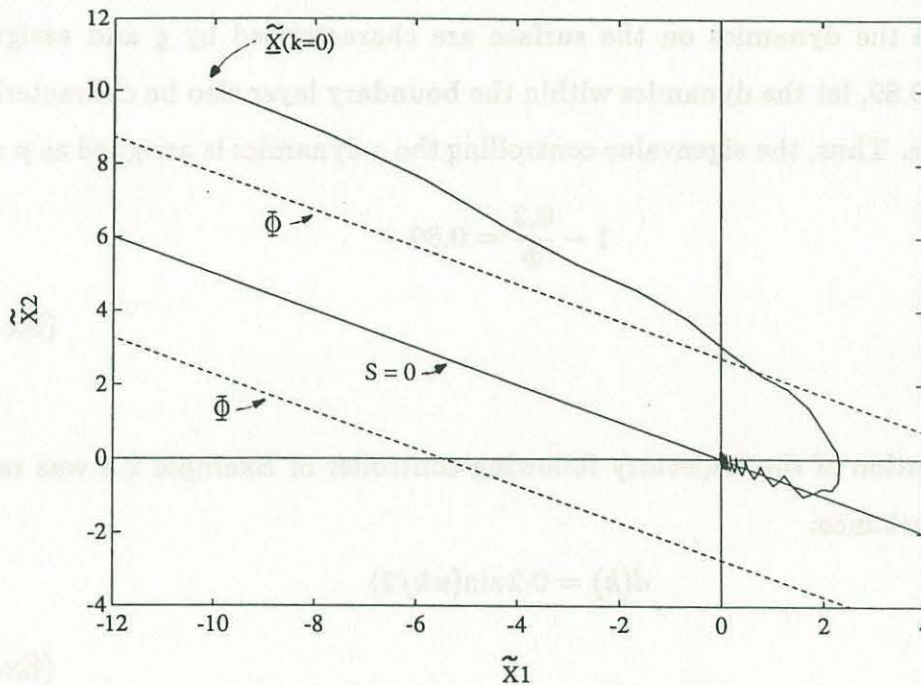


Figure 3.2 Error phase plane

The control input $u(k)$ is shown in Figure 3.3 both when using the smoothed control law and when using the discontinuous control law without boundary layer. As predicted, the smoothed control law is a filtered version of the discontinuous control law, where the filter bandwidth is determined by boundary layer thickness Φ .

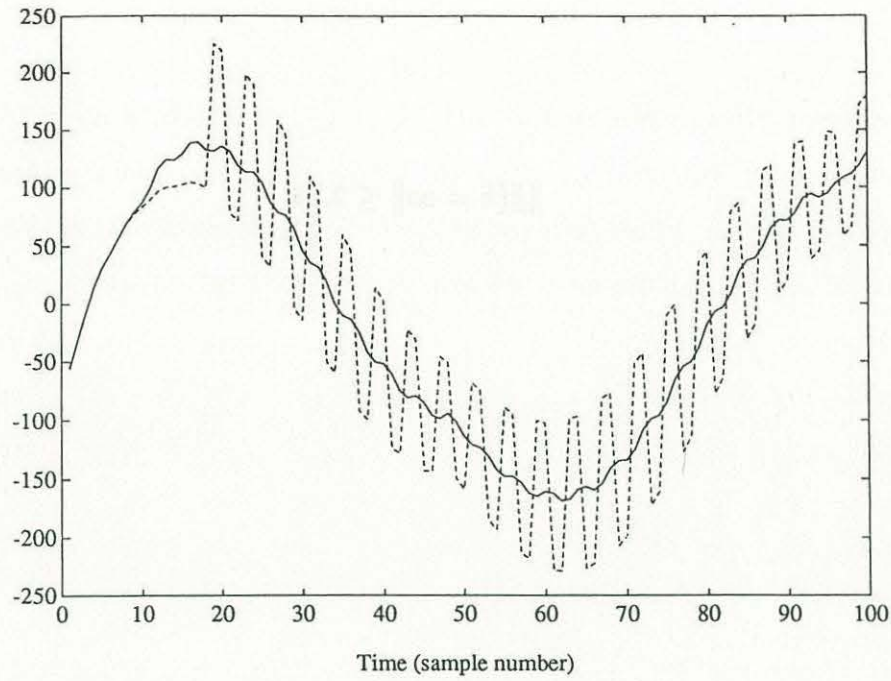


Figure 3.3 Control input using smoothed (-) and discontinuous (- -) control law

To determine the effect of the disturbance on the tracking performance, form the closed loop error system:

$$\begin{bmatrix} \tilde{x}_1(k+1) \\ \tilde{x}_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.22 \\ -0.055 & 0.78 \end{bmatrix} \begin{bmatrix} \tilde{x}_1(k) \\ \tilde{x}_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ d(k) \end{bmatrix} \quad (\text{Ex. 3.1.4})$$

From the discussion in Example 2.1, this implies (using Ex. 2.1.23):

$$\|\tilde{x}(k = \infty)\| \leq \frac{1}{1-p} |d(k)| \quad (\text{Ex. 3.1.5})$$

Now since $d(k)$ is bounded by D which is in turn bounded by K , this implies from (Ex. 3.1.2) that $|d(k)| \leq (1-p)\Phi$. The bound of (Ex. 3.1.5) becomes:

$$\|\tilde{x}(k = \infty)\| \leq \Phi \quad (\text{Ex. 3.1.6})$$

or:

$$\|\tilde{x}(k = \infty)\| \leq 2.73$$

(Ex. 3.1.7)

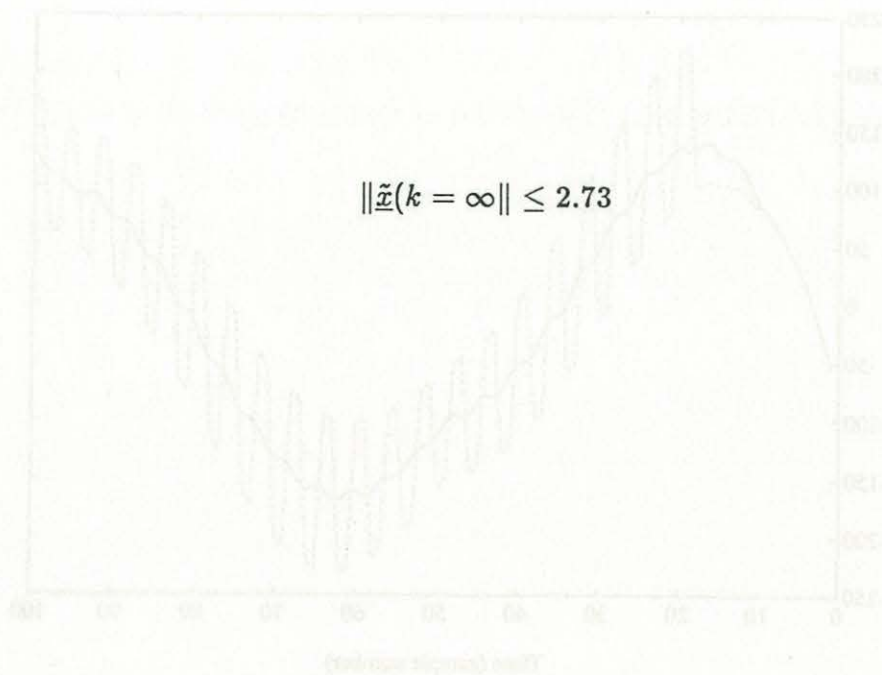


Figure 3.3 Control input $u(k)$ (dashed line) and disturbance $d(k)$ (solid line)

To determine the effect of the disturbance on the tracking performance, form the

closed loop error system

$$\begin{bmatrix} \tilde{x}_1(k+1) \\ \tilde{x}_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.12 \\ -0.055 & 0.12 \end{bmatrix} \begin{bmatrix} \tilde{x}_1(k) \\ \tilde{x}_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ d(k) \end{bmatrix}$$

(Ex. 3.1.4)

From the discussion in Example 3.1, the Lyapunov (using Ex. 3.1.3):

$$\|\tilde{x}(k = \infty)\| \leq \frac{1}{1 - \alpha} \|d(k)\|$$

(Ex. 3.1.5)

Now since $d(k)$ is bounded by \bar{d} which is in turn bounded by K , the input from (Ex.

3.1.2) that $\|d(k)\| \leq (1 - \alpha)\bar{d}$. The bound of (Ex. 3.1.5) becomes

$$\|\tilde{x}(k = \infty)\| \leq \bar{d}$$

(Ex. 3.1.6)

3.2 Uncertainty in System Dynamics

Consider the system of (3.1) where $d(k) = 0$ but with only an estimate for the system dynamics term f available. The system model to the control designer is thus:

$$\begin{aligned}\underline{x}'(k+1) &= \underline{A}'\underline{x}(k) \\ x_n(k+1) &= \hat{f} + bu(k)\end{aligned}\tag{3.2.1}$$

where $\hat{f} = \hat{f}(\underline{x}(k), k)$ and the “^” superscript refers to an estimated quantity. The control law of (3.1.2) becomes:

$$\begin{aligned}u(k) &= \frac{1}{b} \left\{ s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1) + \right. \\ &\quad \left. - K(k) \operatorname{sgn}\{s(k)\} \right\}\end{aligned}\tag{3.2.2}$$

where K has been made time-variant to support the development to follow. The s dynamics from the closed loop system are therefore:

$$s(k+1) = s(k) - K(k) \operatorname{sgn}\{s(k)\} + \tilde{f}\tag{3.2.3}$$

where $\tilde{f} = \tilde{f}(\underline{x}(k), k) = f - \hat{f}$.

It is now necessary to determine an appropriate value for $K(k)$ such that the sliding surface (or a neighboring region) is “attractive” to the system error state for any given initial condition. Formally, this requires showing that $K(k)$ can be selected such that:

$$V(k) = |s(k)| \quad (3.2.4)$$

is a Liapunov function of (3.2.3). Forming the first forward difference of (3.2.4) using (3.23) gives:

$$\Delta V(k) = V(k+1) - V(k) = |s(k) - K(k) \operatorname{sgn}\{s(k)\} + \tilde{f}| - |s(k)| \quad (3.2.5)$$

Now impose the restriction on \tilde{f} that it is boundable by some continuous, linear function of the state $F = F(\underline{x}(k), k)$:

$$F \geq |\tilde{f}| \quad (3.2.6)$$

To make (3.2.5) negative requires two conditions on $K(k)$:

$$K(k) > F \quad (3.2.7)$$

and:

$$K(k) < 2|s(k)| - F \quad (3.2.8)$$

Herein lies one of the major differences between the continuous time and discrete time development alluded to earlier. In the continuous time derivation only one condition on $K(k)$ is imposed so that it is large enough to counteract the uncertainty in system dynamics. This condition is identical in form to that of (3.2.7), and essentially

keeps the s trajectory leading towards the sliding surface. The additional constraint of (3.2.8) in the discrete time case is necessary to keep the s trajectory from travelling from an initial position on one side of the sliding surface to a final position on the other side of the surface (but farther away than before) in one sample period. This can occur because the system acts in an open loop manner between sampling instants.

If $K(k)$ is selected as:

$$K(k) = F + \eta \quad (3.2.9)$$

then (3.2.4) will be a Liapunov function in the region:

$$|s(k)| \geq F + \frac{\eta}{2} \quad (3.2.10)$$

where both conditions on $K(k)$ as given by (3.2.7) and (3.2.8) will be satisfied.

It has therefore been shown that in the region where (3.2.10) holds, the discontinuous control law of (3.2.2) causes the s trajectory to always move closer to the sliding surface in accordance with the Liapunov function of (3.2.4) and can reach this region in finite time using the $K(k)$ of (3.2.9). What remains to be shown is that the closed loop system is stable within this region. The difference between this case and that considered in Section 3.1 is that here s is not converging to a bounded region (given by $|s(k)| < F + \frac{\eta}{2}$) unless F (and thus the state since $F = F(\underline{x}(k), k)$) is bounded.

As in Section 3.1, the control law of (3.2.2) can be modified by introducing a boundary layer around the sliding surface:

$$u(k) = \frac{1}{b} \left\{ s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{A}' \underline{x}_d(k+1) + \right. \\ \left. - \overline{K}(k) \text{sat} \left\{ \frac{s(k)}{\Phi(k)} \right\} \right\} \quad (3.2.11)$$

where $\overline{K}(k)$ and $\Phi(k)$ are to be determined. Note that the boundary layer $\Phi = \Phi(k)$ has now become time-variant in anticipation of utilizing the layer to compensate for the time-variant system dynamics error.

To determine $\overline{K}(k)$, the objective is now to have s converge to the boundary layer rather than the sliding surface itself. The Liapunov function candidate thus becomes:

$$V(k) = | | s(k) | - \Phi(k) | \quad (3.2.12)$$

In a similar manner to (3.2.4)-(3.2.9), the required $\overline{K}(k)$ is given by:

$$\overline{K}(k) = K(k) + \Phi(k) - \Phi(k-1) \quad (3.2.13)$$

with $K(k)$ give by (3.2.9). This guarantees convergence to the boundary layer if initially outside of it. Once inside the boundary layer, the s dynamics become:

$$s(k+1) = s(k) - \frac{\overline{K}(k)}{\Phi(k)} s(k) + \tilde{f} \quad (3.2.14)$$

which mimics a first order filter in s with input \tilde{f} . If the desired eigenvalue of this filter is p , then $\Phi(k)$ must cause:

$$1 - \frac{\overline{K}(k)}{\Phi(k)} = p \quad (3.2.15)$$

Using (3.2.13) in (3.2.15) gives:

$$\Phi(k+1) = p\Phi(k) + K(k) \quad (3.2.16)$$

which specifies the updating law for $\Phi(k)$. The initial condition for (3.2.16) is:

$$\Phi(k=0) = \frac{K(k=0)}{1-p} \quad (3.2.17)$$

The behavior of s inside the boundary layer takes the form:

$$s(k+1) = ps(k) + \tilde{f} \quad (3.2.18)$$

This implies (assuming a low pass filter structure has been selected for s for which $0 < p < 1$):

$$|s(k+1)| \leq p|s(k)| + |\tilde{f}| \quad (3.2.19)$$

Since $|s(k)| \leq \Phi(k)$ inside the boundary layer:

$$|s(k+1)| \leq p\Phi(k) + |\tilde{f}| \quad (3.2.20)$$

By the definition of F from (3.2.6) and $K(k)$ from (3.2.9), (3.2.16) leads to:

$$\Phi(k+1) \geq p\Phi(k) + |\tilde{f}| + \eta \quad (3.2.21)$$

Comparison between (3.2.20) and (3.2.21) reveals that:

$$\Phi(k+1) > |s(k+1)| \quad (3.2.22)$$

which means the s trajectory remains within the boundary layer once inside.

Returning to the stability issue inside the boundary layer, notice that by assigning eigenvalue p to the dynamics of s as in (3.2.15), the control law of (3.2.11) essentially becomes:

$$u(k) = \frac{1}{b} \{ ps(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}^T \underline{x}_d(k+1) \} \quad (3.2.23)$$

and the closed loop error system:

$$\tilde{\underline{x}}(k+1) = \begin{bmatrix} \underline{A}' \\ -\underline{e}'^T \underline{A}' + p\underline{e}^T \end{bmatrix} \tilde{\underline{x}}(k) + \begin{bmatrix} 0 \\ \tilde{f} \end{bmatrix} \quad (3.2.24)$$

Since \tilde{f} has been restricted to a class of functions which is linearly boundable by $F = F(\underline{x}(k), k)$, let:

$$F(\underline{x}(k), k) = | \underline{F}'^T \underline{x}(k) | \quad (3.2.25)$$

where $\underline{F}' = \underline{F}'(k) = [F'_1 \dots F'_n]^T$ whose elements are positive. The system of (3.2.24) can therefore be written:

$$\tilde{\underline{x}}(k+1) = \begin{bmatrix} \underline{A}' \\ -\underline{e}'^T \underline{A}' + p\underline{e}^T \pm \underline{F}'^T \end{bmatrix} \tilde{\underline{x}}(k) + \begin{bmatrix} 0 \\ \pm \underline{F}'^T \underline{x}_d(k) \end{bmatrix} \quad (3.2.26)$$

where the term $\pm \underline{F}'^T$ indicates that any value between $-F'_i$ and $+F'_i$ could be taken for the $i = 1$ to n elements of \underline{F}' . Since $\underline{x}_d(k)$ is bounded, the system stability depends upon the matrix:

$$\begin{bmatrix} A' \\ -\underline{e}^T A' + p\underline{e}^T \pm \underline{F}'^T \end{bmatrix} \quad (3.2.27)$$

having its eigenvalues within the unit circle for all possible values of $\pm \underline{F}'^T$. That is to say, system stability is assured if the matrix is convergent from one sampling instant to the next. This permissible range of elements of \underline{F}' will depend on the bandwidth available as determined by the value of p . In general, a lower bandwidth available will result in a smaller range of permissible values for \underline{F}' . In fact, it may not be possible to show stability for some bandwidth when a certain range of \underline{F}' exists. This is a result of the discrete time application of the control law and the open loop behavior which takes place between sampling instants, as well as the limitations in using a feedback term s which is linear in the error state $\tilde{x}(k)$.

A search procedure can be used to determine the eigenvalues of the matrix (3.2.27) over the (possibly) n dimensional space spanned by \underline{F}' . It is also possible to use an analytical method for simple systems by directly calculating the eigenvalues of (3.2.27) or using a method such as Jury's criteria [71].

The presence of disturbances as well as errors in the system dynamics can be accommodated by simply augmenting $K(k)$ with D as in Section 3.1:

$$K(k) = F + D + \eta \quad (3.2.28)$$

Time-variant disturbances with bound $D = D(k)$ can also be handled by utilizing the time-variant boundary layer as well.

To make the boundary layer determination more immune to noise and disturbances in the measured state vector $\underline{x}(k)$, the update law for $\Phi(k)$ of (3.2.16) can be changed

to:

$$\Phi(k+1) = p\Phi(k) + K_d(k) \quad (3.2.29)$$

where the "d" subscript indicates that the expression is evaluated along the desired trajectory $\underline{x}_d(k)$. Thus:

$$K_d(k) = F(\underline{x}_d(k)) + \eta \quad (3.2.30)$$

Substitution of (3.2.29) and (3.2.30) into (3.2.13) gives the modified expression for $\overline{K}(k)$.

$$\overline{K}(k) = K(k) - K_d(k) + (1-p)\Phi(k) \quad (3.2.31)$$

where $K(k)$ is given by (3.2.9) as before. Although this approach does result in a completely deterministic boundary layer formulation the eigenvalue of the s dynamics becomes time-variant since $\overline{K}(k)/\Phi(k)$ is no longer constant. Thus the stability arguments based on an eigenvalue of p will be approximate rather than strictly true.

Example 3.2

Let the true system be given by (Ex. 2.1.1) but with the available model:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.22 \\ 0 & f' \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{0.0066}{1+0.05|x_2(k)|} \end{bmatrix} u(k) \quad (\text{Ex. 3.2.1})$$

where $0.65 \leq \hat{f} \leq 0.95$. To minimize the magnitude of $K(k)$ by making f as small as possible, select \hat{f} as the algebraic mean of the upper and lower limits of f' :

$$\hat{f} = \frac{\max\{f'\} + \min\{f'\}}{2} = 0.8 \quad (\text{Ex. 3.2.2})$$

The linear boundedness condition of (3.2.6) is therefore met with:

$$F = 0.15 |x_2(k)| \quad (\text{Ex. 3.2.3})$$

Before specifying the details of the control law, first investigate the stability properties of the closed loop system matrix of (3.2.27) for the anticipated bandwidth as set by p . As in Example 3.1, let the eigenvalue controlling the $s(k)$ dynamics around the surface be set equal to the dynamics on the surface at 0.89. The system matrix to be evaluated is thus:

$$\begin{bmatrix} 1 & 0.22 \\ -0.055 & 0.78 \pm F'_2 \end{bmatrix} \quad (\text{Ex. 3.2.4})$$

where $\pm F'_2$ can take on values ranging from -0.15 to 0.15. Figure 3.4 illustrates how the maximum eigenvalue magnitude of (Ex. 3.2.4) varies at F'_2 ranges from -0.3 to 0.3. Since the maximum magnitude of the eigenvalues remain < 1 (i.e. within the unit circle) over the expected range of -0.15 to 0.15 for F'_2 , then the closed loop system stability

is assured inside the boundary layer. It is also clear that for values of F'_2 larger than ≈ 0.2 , the closed loop system would be unstable using the bandwidth corresponding to $p = 0.89$.

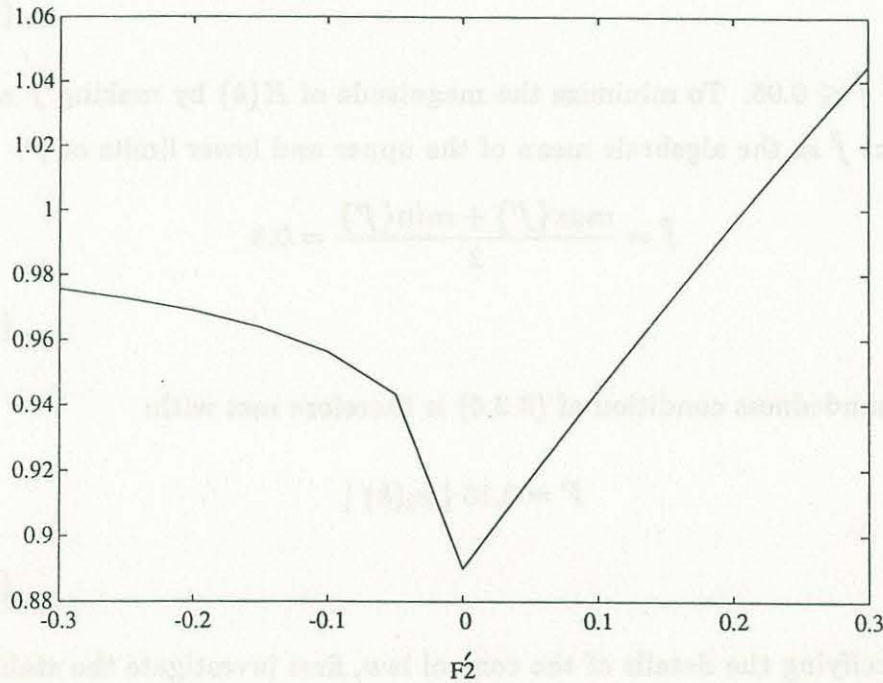


Figure 3.4 Maximum eigenvalue magnitude of closed loop system

It is interesting to consider the value of p necessary to ensure stability over the range of F'_2 values possible. Figure 3.5 shows the effect of various controller bandwidths corresponding to eigenvalues ranging from 0.89 to 0.99 for a range of F'_2 from -0.3 to 0.3. As the eigenvalue gets larger (and the bandwidth lower), the range of error F'_2 which can be accommodated decreases. A value of $p \leq 0.95$ is seen to be necessary to support a range of F'_2 from -0.15 to 0.15.

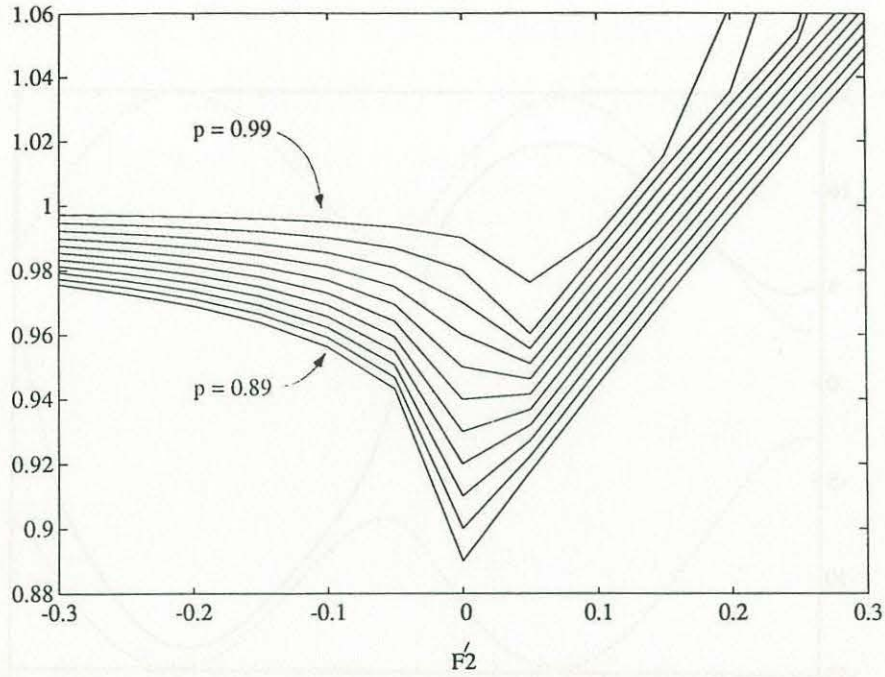


Figure 3.5 Maximum eigenvalue magnitude of closed loop system for different bandwidths

The control law of (3.2.11) is used with $\hat{f} = 0.8x_2(k)$. The boundary layer is determined using (3.2.29) and $K(k)$ by (3.2.28) to accommodate a disturbance identical to that used in Example 3.1 (i.e. bounded by $D = 0.2$). A simulation was performed with the actual value of $f = 0.95x_2(k)$ using the desired trajectory of Example 2.2, and initial condition $\underline{x}(k = 0) = [-10 \ 10]^T$.

The s trajectory is shown in Figure 3.6 along with the boundary layer Φ . As predicted, s converges to the boundary layer and then remains within it. The control input u is shown in Figure 3.7 to demonstrate its bandwidth limited behavior.

As in Example 3.1, the tracking precision is approximately given by:

$$\|\tilde{\underline{x}}(k)\| \leq \Phi(k)$$

(Ex. 3.2.5)

which provides a time-variant precision estimate.

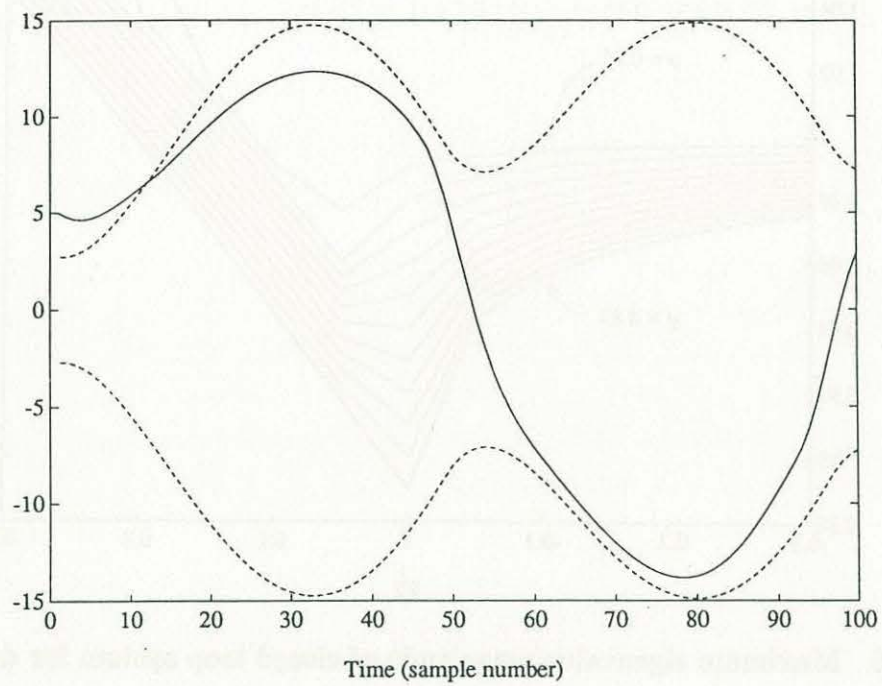


Figure 3.6 S (-) and Φ (- -)

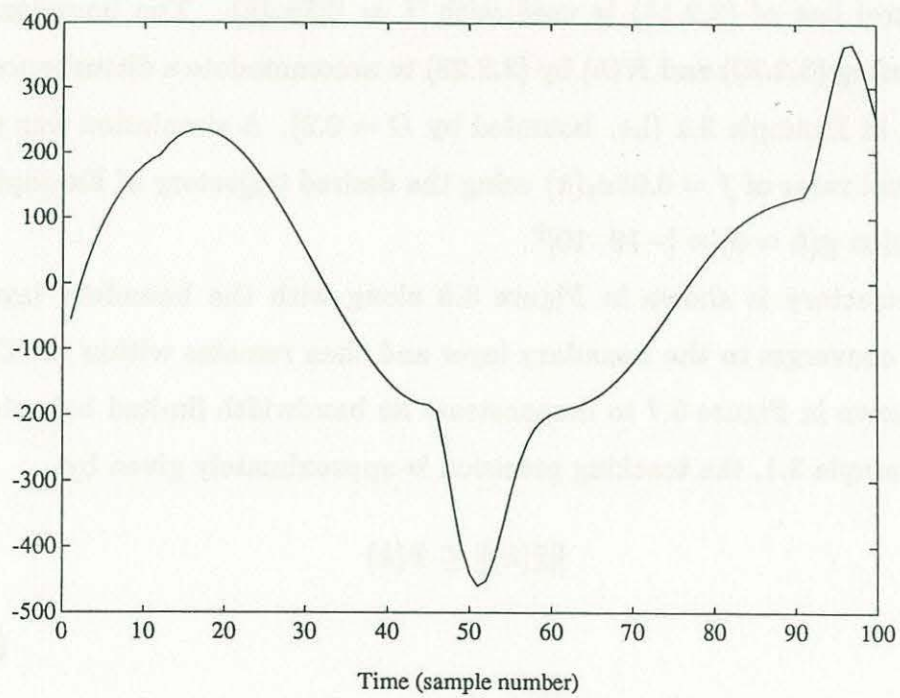


Figure 3.7 Control input

3.3 Uncertainty in Control Gain

Return to the system of (3.1) where $d(k) = 0$ but with only an estimate for the system dynamics term f and control gain term b available. The system model available to the control designer is thus:

$$\begin{aligned}\underline{x}'(k+1) &= \underline{A}'\underline{x}(k) \\ x_n(k+1) &= \hat{f} + \hat{b}u(k)\end{aligned}\tag{3.3.1}$$

where $\hat{f} = \hat{f}(\underline{x}(k), k)$ and $\hat{b} = \hat{b}(\underline{x}(k), k)$. The control law of (3.1.2) becomes:

$$u(k) = \frac{1}{\hat{b}} \left\{ s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1) - K(k) \text{sgn}\{s(k)\} \right\}\tag{3.3.2}$$

The s dynamics from the closed loop system are therefore:

$$\begin{aligned}s(k+1) &= s(k) - \frac{b}{\hat{b}} K(k) \text{sgn}\{s(k)\} + \tilde{f} + \\ &+ \left\{ \frac{b}{\hat{b}} - 1 \right\} \left\{ s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1) \right\}\end{aligned}\tag{3.3.3}$$

where $\tilde{f} = f - \hat{f}$ and is linearly bounded by F as in Section 3.2. Furthermore, let $\frac{b}{\hat{b}}$ be bounded by the linear function $\beta = \beta(\underline{x}(k))$:

$$\beta^{-1} \leq \sqrt{\frac{\max\{b\}}{\min\{b\}}} \leq \beta\tag{3.3.4}$$

where $\max\{b\}$ and $\min\{b\}$ represent the maximum and minimum value that respectively determine the limits of uncertainty on b .

As in Section 3.2, it is necessary to determine an appropriate value for $K(k)$ such that the sliding surface (or a neighboring region) is "attractive" to the system error state for any given initial condition. The Liapunov function of (3.2.4) is again used:

$$V(k) = |s(k)| \quad (3.3.5)$$

which imposes two conditions on $K(k)$:

$$K(k) > \beta F + (\beta - 1) |s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1)| \quad (3.3.6)$$

and:

$$K(k) < \frac{2}{\beta} |s(k)| - \frac{F}{\beta} - \frac{\beta - 1}{\beta} |s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1)| \quad (3.3.7)$$

If $K(k)$ is selected as:

$$K(k) = \beta(F + \eta) + (\beta - 1) |s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1)| \quad (3.3.8)$$

then (3.3.5) will be a Liapunov function in the region:

$$|s(k)| \geq \left(\frac{\beta^2 + 1}{2} \right) F + \frac{(\beta + 1)(\beta - 1)}{2} |s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1)| + \frac{\beta^2 \eta}{2} \quad (3.3.9)$$

where both conditions on $K(k)$ as given by (3.3.6) and (3.3.7) will be satisfied.

Inside the boundary layer the control law of (3.3.2) becomes:

$$u(k) = \frac{1}{\hat{b}} \left\{ s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1) - \overline{K}(k) \text{sat} \left\{ \frac{s(k)}{\Phi(k)} \right\} \right\} \quad (3.3.10)$$

To determine $\overline{K}(k)$ such that the s trajectory converges to the boundary layer when outside, use the Liapunov function candidate:

$$V(k) = ||s(k)| - \Phi(k)| \quad (3.3.11)$$

The condition on $\overline{K}(k)$ depends on whether the boundary layer is expanding ($\Phi(k+1) \geq \Phi(k)$) or contracting ($\Phi(k+1) < \Phi(k)$):

$$\text{If } \Phi(k+1) \geq \Phi(k): \quad \overline{K}(k) = K(k) + \frac{1}{\beta} \{\Phi(k) - \Phi(k+1)\} \quad (3.3.12)$$

$$\text{If } \Phi(k+1) < \Phi(k): \quad \overline{K}(k) = K(k) + \beta \{\Phi(k) - \Phi(k+1)\} \quad (3.3.13)$$

where $K(k)$ is given by (3.3.8).

Inside the boundary layer, the s dynamics are:

$$s(k+1) = s(k) - \frac{b\overline{K}(k)}{\hat{b}\Phi(k)} + \tilde{f} + \left\{ \frac{b}{\hat{b}} - 1 \right\} \left\{ s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1) \right\} \quad (3.3.14)$$

which mimics a first order filter in s with input

$$\tilde{f} + \left\{ \frac{b}{\hat{b}} - 1 \right\} \left\{ s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1) \right\}.$$

If the desired eigenvalue of this filter is p , then $\Phi(k)$ must cause:

$$1 - \frac{b \bar{K}(k)}{\hat{b} \Phi(k)} = p$$

(3.3.15)

From (3.3.15), it is clear that the filter bandwidth can not be set exactly because b is not known. Assuming that a stable low pass filter structure has been selected which implies $0 < p < 1$, then $\Phi(k)$ should be selected such that the actual eigenvalue of the filter lies between p and 1. This can be accomplished by letting the update law for $\Phi(k)$ be:

$$\text{If } K(k) \geq \frac{1-p}{\beta} \Phi(k) : \Phi(k+1) = p\Phi(k) + \beta K(k)$$

(3.3.16)

$$\text{If } K(k) \geq \frac{1-p}{\beta} \Phi(k) : \Phi(k+1) = \frac{\beta^2 - 1 + p}{\beta^2} \Phi(k) + \frac{K(k)}{\beta}$$

(3.3.17)

where (3.3.16) corresponds to boundary layer expansion and is thus used with the $\bar{K}(k)$ of (3.3.12), whereas (3.3.17) corresponds to boundary layer contraction and thus is used with the $\bar{K}(k)$ of (3.3.13). In either case, the eigenvalue characterizing the filter structure of (3.3.15) ranges from p to $\frac{\beta^2 - 1 + p}{\beta^2}$ (which is < 1) as required. The initial condition used for $\Phi(k)$ is:

$$\Phi(k=0) = \frac{\beta(x(k=0))K(k=0)}{1-p}$$

(3.3.18)

as obtained from (3.3.16) or (3.3.17).

The behavior of s inside the boundary layer takes the form:

$$s(k+1) = \gamma s(k) + \tilde{f} + \left\{ \frac{b}{\hat{b}} - 1 \right\} \left\{ s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1) \right\} \quad (3.3.19)$$

where $p \leq \gamma < \frac{\beta^2 - 1 + p}{\beta^2}$. This implies:

$$|s(k+1)| \leq \frac{\beta^2 - 1 + p}{\beta^2} |s(k)| + |\tilde{f}| + \left| \frac{b}{\hat{b}} - 1 \right| |s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1)| \quad (3.3.20)$$

Since $|s(k)| < \Phi(k)$ inside the boundary layer:

$$|s(k+1)| < \frac{\beta^2 - 1 + p}{\beta^2} \Phi(k) + |\tilde{f}| + \left| \frac{b}{\hat{b}} - 1 \right| |s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1)| \quad (3.3.21)$$

Using the updating law for $\Phi(k)$ for boundary layer contraction (3.3.17) and the $K(k)$ of (3.3.8) gives:

$$\Phi(k+1) \geq \frac{\beta^2 - 1 + p}{\beta^2} \Phi(k) + |\tilde{f}| + \left| \frac{b}{\hat{b}} - 1 \right| |s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1)| + \eta \quad (3.3.22)$$

and comparison between (3.3.21) and (3.3.22) shows that $\Phi(k+1) > |s(k+1)|$ during boundary layer contraction. For the case of boundary layer expansion, using the updating law for $\Phi(k)$ of (3.3.16) and the $K(k)$ of (3.3.8) gives:

$$\Phi(k+1) \geq p\Phi(k) + |\tilde{f}| + \left| \frac{b}{\hat{b}} - 1 \right| |s(k) - \hat{f} - \underline{e}'^T \underline{A}' \underline{x}(k) + \underline{e}'^T \underline{x}_d(k+1)| + \eta + \left(\beta - \frac{1}{\beta}\right) K(k) \quad (3.3.23)$$

But since $K(k) \geq \frac{1-p}{\beta} \Phi(k)$ during boundary layer expansion (from 3.3.16), (3.3.23) becomes:

$$\Phi(k+1) \geq \frac{\beta^2 - 1 + p}{\beta^2} \Phi(k) + |\tilde{f}| + \left| \frac{b}{\hat{b}} - 1 \right| \|s(k) - \hat{f} - \underline{e}^T \underline{A}' \underline{x}(k) + \underline{e}^T \underline{x}_d(k+1)\| + \eta \quad (3.3.24)$$

which, as before, bounds $|s(k+1)|$. As in section 3.2, this means that the s trajectory remains within the boundary layer once inside.

To determine the stability inside the boundary layer, first note that by the selection of $\bar{K}(k)$ and $\Phi(k)$, the control law of (3.3.10) essentially becomes:

$$u(k) = \frac{1}{\hat{b}} \left\{ \frac{\beta - 1 + p}{\beta} s(k) - \hat{f} - \underline{e}^T \underline{A}' \underline{x}(k) + \underline{e}^T \underline{x}_d(k+1) \right\} \quad (3.3.25)$$

making the closed loop error system:

$$\tilde{\underline{x}}(k+1) = \begin{bmatrix} \underline{A}' \\ \frac{b}{\hat{b}} \left(-\underline{e}^T \underline{A}' + \frac{\beta-1+p}{\beta} \underline{e}^T \right) \end{bmatrix} \tilde{\underline{x}}(k) + \begin{bmatrix} 0 \\ \tilde{f} + \left\{ \frac{b}{\hat{b}} - 1 \right\} \{ -\hat{f} - \underline{e}^T \underline{x}_d(k+1) \} \end{bmatrix} \quad (3.3.26)$$

Let \tilde{f} be linearly bounded by:

$$|\underline{F}'''^T \underline{x}(k)| \geq |\hat{f}| \quad (3.3.27)$$

where $\underline{F}''' = [F_1''' \dots F_n''']^T$ whose elements are constant elements. Further manipulation of (3.3.26) using (3.3.27) and using the fact that \tilde{f} is bounded by F (and thus by $\underline{F}' \|\underline{x}(k)\|$) as in Section 3.2 and $\frac{b}{\hat{b}}$ by β , then (3.3.27) becomes:

$$\begin{aligned} \tilde{\underline{x}}(k+1) = & \begin{bmatrix} \underline{A}' \\ \beta' \left(-\underline{e}^T \underline{A}' + \frac{\beta-1+p}{\beta} \underline{e}^T - \underline{F}''' \right) \pm \underline{F}' + \underline{F}'' \end{bmatrix} \tilde{\underline{x}}(k) + \\ & + \begin{bmatrix} 0 \\ \pm \underline{F}' \underline{x}_d(k) - \{\beta' - 1\} \{ \underline{F}'' \underline{x}_d(k) + \underline{e}^T \underline{x}_d(k+1) \} \end{bmatrix} \end{aligned} \quad (3.3.28)$$

where $\pm F'$ and β' indicates that any value between $-F'_i$ and $+F'_i$ and $\frac{1}{\beta}$ and β , respectively, could be taken for the $i = 1$ to n elements of \underline{F}' . Since $\underline{x}_d(k)$ is bounded, the system stability depends on the matrix:

$$\begin{bmatrix} \underline{A}' \\ \beta' \left(-\underline{e}^T \underline{A}' + \frac{\beta-1+p}{\beta} \underline{e}^T - \underline{F}'' \right) \pm \underline{F}' + \underline{F}'' \end{bmatrix} \quad (3.3.29)$$

having its eigenvalues within the unit circle for all possible values of $\pm \underline{F}'$ and $\pm \beta$. This is only a rigorous stability proof if \hat{f} is a linear function of the state $\underline{x}(k)$, however, due to the necessity of using the bound \underline{F}' .

As explained in Section 3.2, the presence of disturbances can be accommodated by augmenting the $K(k)$ of (3.3.8) with the disturbance bound D :

$$K(k) = \beta(F + D + \eta) + (\beta - 1) | s(k) - \hat{f} - \underline{e}^T \underline{A}' \underline{x}(k) + \underline{e}^T \underline{x}_d(k+1) | \quad (3.3.30)$$

The boundary layer can be calculated using the desired trajectory $\underline{x}_d(k)$ to increase the immunity to noise and disturbances. The updating laws for $\Phi(k)$ of (3.3.16) and (3.3.17) become:

$$\text{If } K_d(k) \geq \frac{1-p}{\beta} \Phi(k) : \quad \Phi(k+1) = p\Phi(k) + \beta K_d(k) \quad (3.3.31)$$

$$\text{If } K_d(k) < \frac{1-p}{\beta} \Phi(k) : \quad \Phi(k+1) = \frac{\beta^2 - 1 + p}{\beta^2} \Phi(k) + \frac{K_d(k)}{\beta} \quad (3.3.32)$$

where $K_d(k)$ is $K(k)$ evaluated along $\underline{x}_d(k)$. Using (3.3.31) and (3.3.32), $\bar{K}(k)$ is calculated as:

$$\text{If } \Phi(k+1) \geq \Phi(k) : \bar{K}(k) = K(k) - K_d(k) + \frac{1}{\beta}(1-p)\Phi(k) \quad (3.3.33)$$

$$\text{If } \Phi(k+1) < \Phi(k) : \bar{K}(k) = K(k) - K_d(k) + \beta(1-p)\Phi(k) \quad (3.3.34)$$

and the initial condition for $\Phi(k)$ becomes:

$$\Phi(k=0) = \frac{\beta(x_d(k=0))K_d(k=0)}{1-p} \quad (3.3.35)$$

Example 3.3

Let the true system be given by (Ex. 2.1.1) but with the available model:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.22 \\ 0 & f' \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{b'}{1+0.05|x_2(k)|} \end{bmatrix} u(k) \quad (\text{Ex. 3.3.1})$$

where $0.65 \leq f' \leq 0.95$ and $0.0066 \leq b' \leq 0.0103$. As in Example 3.2, \hat{f} is selected as the algebraic mean of f' , or 0.8. In a similar manner to minimize the magnitude of $K(k)$ by making β as small as possible, select \hat{b} as the geometric mean of the upper and lower limits of b' :

$$\hat{b} = \sqrt{\max\{b'\} \cdot \min\{b'\}} = 0.0082 \quad (\text{Ex. 3.3.2})$$

Using (3.3.4) to compute β :

$$\beta = \sqrt{\frac{\max\{b'\}}{\min\{b'\}}} = 1.25 \quad (\text{Ex. 3.3.3})$$

The closed loop system matrix of (3.3.29) must be evaluated for stability inside the boundary layer, and has the form:

$$\begin{bmatrix} 1 & 0.22 \\ -0.044\beta' & 0.002\beta' + 0.8 \pm F'_2 \end{bmatrix} \quad (\text{Ex. 3.3.4})$$

where $\pm F'_2$ can take on values ranging from -0.15 to 0.15 and β' from 0.8 to 1.25. Figure 3.8 illustrates how the maximum eigenvalue magnitude of (Ex. 3.3.4) varies with values of F'_2 ranging from -0.3 to 0.3 and β' from 0.6 to 2.0. The eigenvalue magnitudes are presented as contours ranging from 0.91 to 1.0. Any region outside the 1.0 contour

represents an unstable system. Any combination of F'_2 and β' in the expected range is seen to be stable.

The control law of (3.3.10) is used with $\hat{f} = 0.8x_2(k)$ and $\hat{b} = 0.0082/(1 + 0.05 |x_2(k)|)$. The boundary layer is determined using (3.3.31) and (3.3.32), and the $K(k)$ of (3.3.3) to accommodate a disturbance identical to that used in Example (3.1) which was bounded by $D=0.2$. A simulation was performed with the actual value of $f = 0.95 x_2(k)$ and $b = 0.0066/(1 + 0.05 |x_2(k)|)$ using the desired trajectory of Example 2.2 and initial condition $\underline{x}(k=0) = [-10 \ 10]^T$.

The s trajectory is shown in Figure 3.9 along with the boundary layer, and is seen to converge to the boundary layer from its initial position outside the layer and remain within it thereafter despite the large error in estimated control gain. The control input shown in Figure 3.10 retains its bandwidth limited behavior.

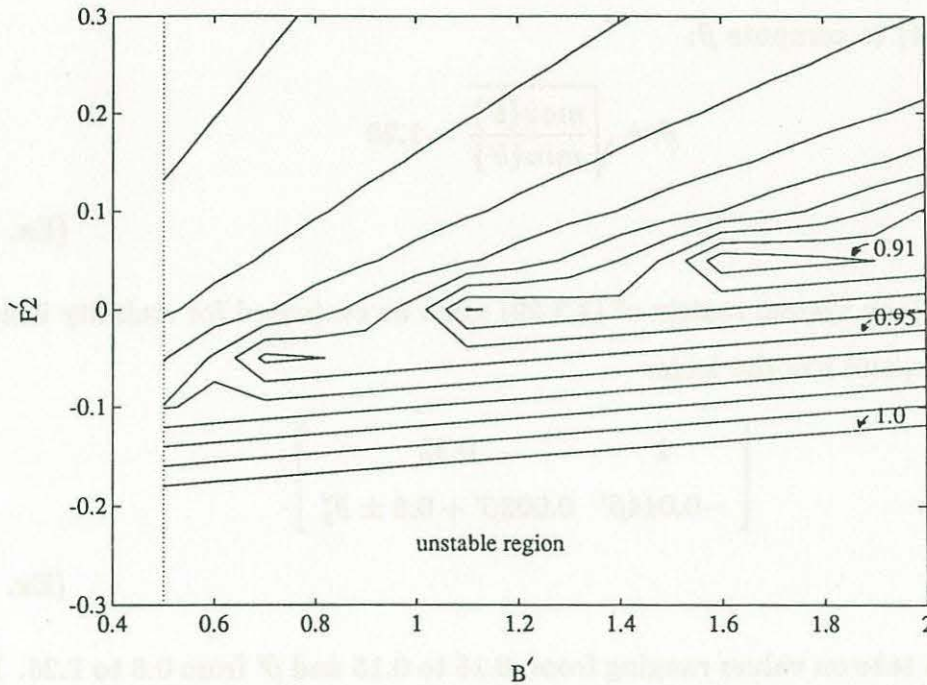


Figure 3.8 Contour plot of maximum eigenvalue magnitude

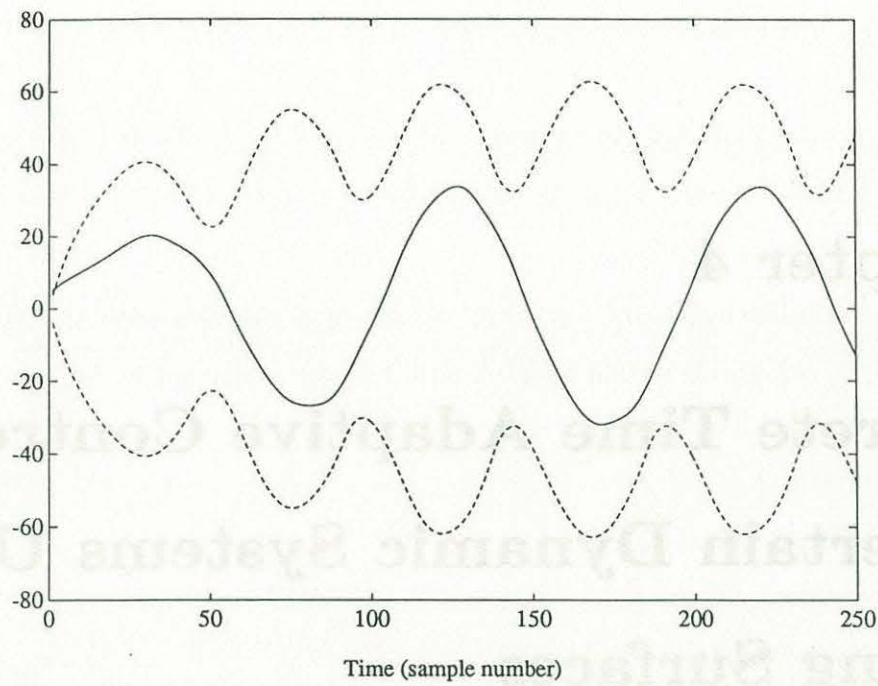


Figure 3.9 S (-) and Φ (- -)

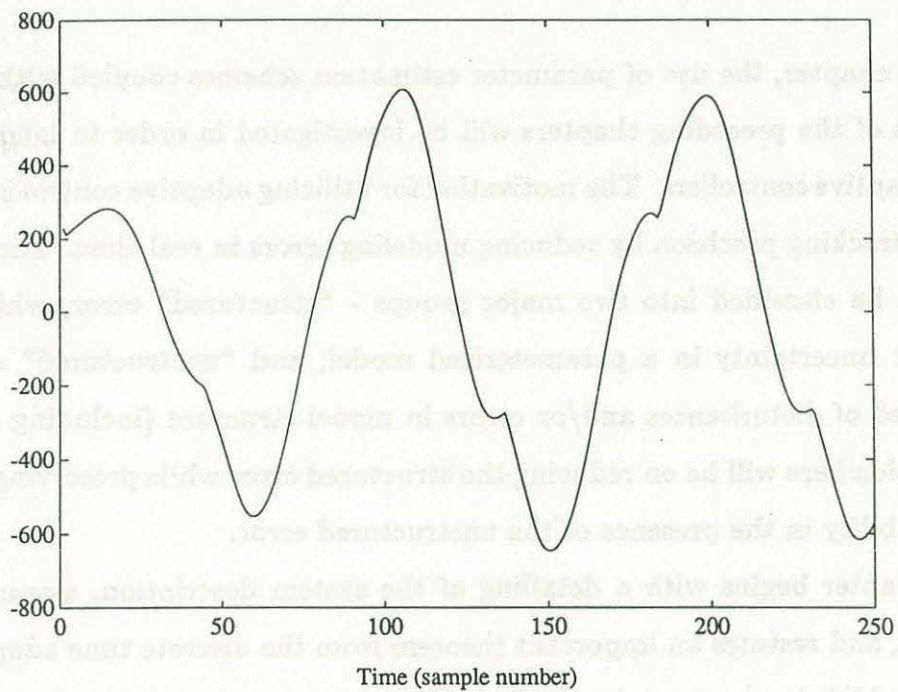


Figure 3.10 Control input

Chapter 4

Discrete Time Adaptive Control of Uncertain Dynamic Systems Using Sliding Surfaces

In this chapter, the use of parameter estimation schemes coupled with the control algorithms of the preceding chapters will be investigated in order to introduce a new class of adaptive controllers. The motivation for utilizing adaptive control is to allow for increased tracking precision by reducing modeling errors in real time. These modeling errors can be classified into two major groups - “structured” error, which refers to parametric uncertainty in a parameterized model, and “unstructured” error, which is composed of disturbances and/or errors in model structure (including order). The concentration here will be on reducing the structured error while preserving closed loop system stability in the presence of the unstructured error.

The chapter begins with a detailing of the system description, assumptions, and definitions, and restates an important theorem from the discrete time adaptive control literature which is necessary in the later development. A variety of parameter estimation algorithms and their characteristics are then presented along with extensions

to constrained parameter estimation and parameter estimation with bounded disturbances. The adaptive controller development for systems with no unmodeled dynamics follows, both for no disturbances and an extension to the bounded disturbance case. Adaptive control in the presence of unmodeled dynamics is then explored, first by investigating possible instabilities that can result and finally by proposing methods to guard against these instabilities.

$$\dot{x}(t) = A x(t) + B u(t)$$

$$x(0) = x_0, \quad u(t) = 0$$

(4.1.1.1)

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times n}$ is a constant matrix, $B \in \mathbb{R}^{n \times m}$ is a constant matrix, and $x_0 \in \mathbb{R}^n$ is a constant vector. The system (4.1.1.1) is a linear time-invariant system with no unmodeled dynamics.

$$\dot{x}(t) = A x(t) + B u(t)$$

$$x(0) = x_0, \quad u(t) = 0$$

(4.1.1.2)

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times n}$ is a constant matrix, $B \in \mathbb{R}^{n \times m}$ is a constant matrix, and $x_0 \in \mathbb{R}^n$ is a constant vector. The system (4.1.1.2) is a linear time-invariant system with no unmodeled dynamics.

As in the preceding chapter, it is assumed that the full state $x(t)$ is known at all times. The system (4.1.1.1) is a linear time-invariant system with no unmodeled dynamics.

The system (4.1.1.2) is a linear time-invariant system with no unmodeled dynamics.

4.1 Preliminary Development

4.1.1 System Description, Assumptions, and Definitions

The use of the adaptive controller is motivated by the presence of uncertainty in the system description. As in Chapter 3, the system form of (3.1) is used:

$$\begin{aligned}\underline{x}'(k+1) &= \underline{A}'\underline{x}(k) \\ x_n(k+1) &= f + bu(k) + d(k)\end{aligned}\tag{4.1.1.1}$$

where $\underline{x}'(k) = [x_1(k) \cdots x_{n-1}(k)]^T$, \underline{A}' is a constant $(n-1) \times n$ dimension matrix, $f = f(\underline{x}(k), k)$ and $b = b(\underline{x}(k), k)$ are scalar functions, and $d(k)$ is a scalar disturbance. Similar to Section 2.6, input time delays can be included if the system can be expressed in the form:

$$\begin{aligned}\underline{x}'(k+d) &= \underline{A}'\underline{x}(k) \\ x_n(k+d) &= f + \sum_{i=1}^d \{b_i z^{-(i-1)}\} u(k) + d(k)\end{aligned}\tag{4.1.1.2}$$

where each $b_i = b_i(\underline{x}(k), k)$ is a scalar function, z is the unit delay operator, and d an integer ≥ 1 representing the input time delay. This more general form will be used for the remainder of the development and reduces to (4.1.1.1) by setting delay $d = 1$.

As in the preceeding chapters, it is assumed that the full state $\underline{x}(k)$ is known at sampling instant k and that the system is completely controllable from input $u(k)$. The adaptive controller development requires the following additional assumptions:

1. The system can be expressed in a form which is linear in a suitably chosed parameter set:

$$\begin{aligned}\underline{x}'(k+d) &= \underline{A}'x(k) \\ x_n(k+d) &= \underline{\Theta}_f^T \underline{\phi}_f + \sum_{i=1}^d \left\{ \underline{\Theta}_{b_i}^T \underline{\phi}_{b_i} z^{-(i-1)} \right\} u(k) + d(k)\end{aligned}\tag{4.1.1.3}$$

where $\underline{\Theta}_f = \underline{\Theta}_f(k)$ is an ℓ dimensional parameter vector, $\underline{\phi}_f = \underline{\phi}_f(\underline{x}(k), k)$ an ℓ dimensional vector, each $\underline{\Theta}_{b_i} = \underline{\Theta}_{b_i}(k)$ an m dimensional parameter vector, and each $\underline{\phi}_{b_i} = \underline{\phi}_{b_i}(\underline{x}(k), k)$ an m dimensional vector.

2. The order of the system model is equal to that of the actual system.
3. The polynomial in z :

$$\sum_{i=1}^d \left\{ \underline{\Theta}_{b_i}^T \underline{\phi}_{b_i} z^{-(i-1)} \right\}\tag{4.1.1.4}$$

has a stable inverse.

4. Each element of $\underline{\phi}_f$ can be linearly bounded by $\underline{x}(k)$ and each element of $\underline{\phi}_{b_i}$ can be linearly bounded by a constant over the n dimensional state space (see Condition (3) in Section 4.1.2 for the definition of linear boundedness).

Assumption 1 allows for the extensive theory of linear parameter estimation to be applied, noting that while (4.1.1.3) is linear in the parameter vectors $\underline{\Theta}_f$ and $\underline{\Theta}_{b_i}$, the regressor vectors $\underline{\phi}_f$ and $\underline{\phi}_{b_i}$ can be nonlinear functions of the state $\underline{x}(k)$ and/or time-variant. Assumption 2 allows the potential for exact parametric matching between the model and the actual system. This assumption is necessary to provide analytic convergence proofs during an idealized analysis, but rarely is strictly applicable in practice. The latter part of this chapter seeks to allow for a partial relaxation of this restriction. Assumption 3 will be required when the adaptive control law is introduced to demonstrate that the control law remains bounded when input time delays are

present. Assumption 4 is used in the adaptive controller boundedness arguments and restricts either the class of nonlinear functions of $\underline{x}(k)$ used in $\underline{\phi}_f$ and $\underline{\phi}_{b_i}$ or the region on the state space where the boundedness arguments will hold.

Much work has been done in the parameter estimation and adaptive control field using a single-input, single-output system description. As pointed out in Chapter 1, a contribution of this research work is a straightforward method for utilizing the additional output information (full state feedback) when it is available. The simplicity achieved when scalar quantities are used in the parameter estimation and control algorithms again motivates the use of the summarizing scalar concept as introduced in Chapter 2.

Using the parameterized but multi-dimensional system (4.1.1.3), a scalar system description can be obtained by pre-multiplying (4.1.1.3) by \underline{e}^T :

$$\underline{e}^T \underline{x}(k+d) = \underline{e}^T \underline{A}' \underline{x}(k) + \underline{\Theta}_f^T \underline{\phi}_f + \sum_{i=1}^d \left\{ \underline{\Theta}_{b_i}^T \underline{\phi}_{b_i} z^{-(i-1)} \right\} u(k) + d(k) \quad (4.1.1.5)$$

where $\underline{e}' = [e_1 \cdots e_{n-1}]^T$. To further simplify the system description and to place it in a format suitable for parameter estimation, introduce the following notation:

$$\underline{\Theta}^T(k) = [\underline{\Theta}_f^T(k) \quad \underline{\Theta}_{b_1}(k) \cdots \underline{\Theta}_{b_d}(k)] \quad (4.1.1.6)$$

$$\underline{\phi}^T(k) = [\underline{\phi}_f^T(k) \quad \underline{\phi}_{b_1}^T(k)u(k) \cdots \underline{\phi}_{b_d}^T(k)u(k-d+1)] \quad (4.1.1.7)$$

where $\underline{\Theta}(k)$ and $\underline{\phi}(k)$ are of dimension $\ell + md$. Using (4.1.1.6) and (4.1.1.7), (4.1.1.5) becomes:

$$\underline{e}^T \underline{x}(k+d) = \underline{e}^T \underline{A}' \underline{x}(k) + \underline{\Theta}^T(k) \underline{\phi}(k) + d(k) \quad (4.1.1.8)$$

In the event that the parameter vector $\underline{\Theta}(k)$ is, in fact, time-variant, it is generally not possible to estimate this vector exactly unless detailed knowledge is available regarding the nature of the time variation. Although comments will be provided later regarding systems with time-variant parameters, the remainder of the development will assume a time-invariant parameter vector ($\underline{\Theta}(k) = \underline{\Theta}$ in (4.1.1.8)).

In the parameter estimation problem, the system output $\underline{e}^T \underline{x}(k+d)$ is estimated using an estimated parameter vector which is time-variant:

$$\underline{e}^T \hat{\underline{x}}(k+d) = \underline{e}^T \underline{A}' \underline{x}(k) + \hat{\underline{\Theta}}^T(k) \underline{\phi}(k) \quad (4.1.1.9)$$

where the implicit assumption in (4.1.1.9) is that $d(k)$ is a zero-mean disturbance. Subtracting (4.1.1.9) from (4.1.1.8) defines the "prediction error":

$$\underline{e}^T \hat{\underline{x}}(k+d) = \tilde{\underline{\Theta}}^T(k) \underline{\phi}(k) + d(k) \quad (4.1.1.10)$$

where $\hat{\underline{x}}(k+d) = \underline{x}(k+d) - \hat{\underline{x}}(k+d)$ and $\tilde{\underline{\Theta}}(k) = \underline{\Theta} - \hat{\underline{\Theta}}(k)$.

A somewhat arbitrary decision is made that separates the state vector $\underline{x}(k)$ and the estimated parameter vector $\hat{\underline{\Theta}}(k)$, since both technically comprise the total "state" of the system. The separation is typically made based on the fact that $\hat{\underline{\Theta}}$ generally converges to a time invariant vector $\underline{\Theta}$, whereas this is in general not the case with $\underline{x}(k)$. Despite the terminology, it should be remembered that any analysis which seeks to demonstrate the boundedness properties of the entire system must take into account both $\underline{x}(k)$ and $\hat{\underline{\Theta}}(k)$.

4.1.2 A Discrete Time Stability Theorem

Using the system description as defined in Section 4.1.1, standard parameter estimation schemes may be used to estimate the parameter vector $\underline{\Theta}$. It must be remembered

that most estimation schemes have an implicit *a priori* assumption that the components of the regressor vector $\underline{\phi}(k)$ are bounded. The estimation stability problem is then one of insuring that the parameter estimates are bounded and converge to some limit. This result is naturally required in the adaptive controller (which utilizes these parameter estimates) in order that the control input remains bounded and that the tracking error converges to some limit. It is further necessary to show, however, that the system state (included in the components of the regressor vector) remain bounded when using a particular control law.

The “Key Technical Lemma” as presented in [72] provides a method that allows for the integration of parameter estimation and adaptive control in discrete time systems. This lemma shows that if a parameter estimation scheme is used which guarantees convergence of the prediction error under certain conditions, a properly chosen adaptive control law which uses this estimation scheme will result in convergence of the tracking error to some limit while keeping the regressor vector bounded. The lemma is repeated here without proof:

If the following conditions are satisfied for some given sequences $p(k), \sigma(k), b_1(k)$, and $b_2(k)$:

1. $\lim_{k \rightarrow \infty} \frac{p^2(k)}{b_1(k) + b_2(k) \underline{\sigma}^T(k) \underline{\sigma}(k)} = 0$
 where $b_1(k), b_2(k)$, and $p(k)$ are real scalar sequences and $\underline{\sigma}(k)$ is a real $n \times 1$ vector sequence.
2. Uniform boundedness condition
 $0 < b_1(k) < K < \infty$ and $0 < b_2(k) < K < \infty$ for all $k \geq 1$
3. Linear boundedness condition
 $\|\underline{\sigma}(k)\| \leq c_1 + c_2 \max\{|p(\tau)|\}$ for $0 \leq \tau \leq k$ where $0 < c_1 < \infty$ and $0 < c_2 < \infty$

Then:

- (i) $\lim_{k \rightarrow \infty} p(k) = 0$
- (ii) $\underline{\sigma}(k)$ is bounded

The development of the adaptive controller will proceed as follows: standard parameter estimation schemes will be presented that apply directly to the scalar system of (4.1.1.8) and that result in bounded parameter estimates and a normalized prediction error (to be defined in Section 4.2) which converges to zero. These properties will be shown without any *a priori* assumptions on the boundedness of the regressor vector $\underline{\phi}(k)$. A control law that utilizes the estimated parameter vector will then be proposed (closely related to the control laws of the preceding chapters) to which the Key Technical Lemma will be applied in order to show boundedness of the regressor vector $\underline{\phi}(k)$ and convergence of the tracking error to some limit.

4.2 Parameter Estimation Methods

The normalized gradient and least squares parameter estimation methods (and some variants) are presented in this section using the system model of (4.1.1.8) with no disturbance ($d(k) = 0$) and no known *a priori* constraints on $\hat{\underline{\Theta}}(k)$. As remarked in the previous section, no assumptions are made on the boundedness of the regressor vector $\underline{\phi}(k)$. The goal in each method is to show that the parameter estimates remain bounded and that the normalized prediction error converges to zero. The need for these properties will become clear when the adaptive control laws are introduced.

4.2.1 Normalized Gradient Parameter Estimation

The normalized gradient parameter estimation algorithm applied to the system of (4.1.1.8) and estimated system (4.1.1.9) is:

$$\hat{\underline{\Theta}}(k) = \hat{\underline{\Theta}}(k-1) + \frac{a(k)\underline{\phi}(k-d)[\underline{e}^T \underline{x}(k) - \underline{e}'^T \underline{A}' \underline{x}(k-d) - \hat{\underline{\Theta}}^T(k-1)\underline{\phi}(k-d)]}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} \quad (4.2.1.1)$$

where $k \geq d$, $0 < a(k) < 2$, and $c > 0$, and the "normalization" refers to the presence of $\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c$ in the denominator of (4.2.1.1).

To show that the parameter estimates are bounded, use the Liapunov function candidate:

$$V(k) = \|\tilde{\underline{\Theta}}(k)\|^2 \quad (4.2.1.2)$$

Using the definition of $\tilde{\underline{\Theta}}(k)$ and (4.2.1.1):

$$\tilde{\mathbf{Q}}(k) = \tilde{\mathbf{Q}}(k-1) - \frac{a(k)\underline{\phi}(k-d)[\underline{e}^T \underline{x}(k) - \underline{e}^T \underline{A}' \underline{x}(k-d) - \hat{\mathbf{Q}}^T(k-1)\underline{\phi}(k-d)]}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} \quad (4.2.1.3)$$

Calculate the first backward difference of the Liapunov function candidate (4.2.1.2):

$$\Delta V(k) = V(k) - V(k-1) = \|\tilde{\mathbf{Q}}(k)\|^2 - \|\tilde{\mathbf{Q}}(k-1)\|^2 \quad (4.2.1.4)$$

Substitution of (4.2.1.3) into (4.2.1.4) gives:

$$\begin{aligned} \Delta V(k) = & \frac{a(k)[\underline{e}^T \underline{x}(k) - \underline{e}^T \underline{A}' \underline{x}(k-d) - \hat{\mathbf{Q}}^T(k-1)\underline{\phi}(k-d)]^2}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} \\ & \cdot \left\{ \frac{a(k)\underline{\phi}^T(k-d)\underline{\phi}(k-d)}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} - 2 \right\} \end{aligned} \quad (4.2.1.5)$$

Since $0 < a(k) < 2$, it is clear from (4.2.1.5) that $\Delta V(k)$ is non-positive. Further, (4.2.1.2) has a unique minimum at $\tilde{\mathbf{Q}}(k) = \underline{0}$ and is always non-negative. Thus, $V(k)$ is a Liapunov function of the estimator system (4.1.1.8) and (4.2.1.1), and thus the parameter estimates will remain bounded.

To show that the normalized prediction error converges to zero, evaluate $\|\tilde{\mathbf{Q}}(k)\|^2$ from $k = 0$ to $k = k$ using (4.2.1.3) and (4.1.1.8):

$$\|\tilde{\mathbf{Q}}(k)\|^2 = \|\tilde{\mathbf{Q}}(k=0)\|^2 + \sum_{i=1}^k \left\{ \left(\frac{a(i)[\tilde{\mathbf{Q}}^T(i-1)\underline{\phi}(i-d)]^2}{\underline{\phi}^T(i-d)\underline{\phi}(i-d) + c} \right) \left(\frac{a(i)\underline{\phi}^T(i-d)\underline{\phi}(i-d)}{\underline{\phi}^T(i-d)\underline{\phi}(i-d) + c} - 2 \right) \right\} \quad (4.2.1.6)$$

Using the fact that $\|\tilde{\mathbf{Q}}(k)\|^2$ is non-negative and bounded and that $0 < a(k) < 2$ implies:

$$\lim_{k \rightarrow \infty} \sum_{i=1}^k \left\{ \frac{[\tilde{\hat{\Theta}}^T(i-1)\underline{\phi}(i-d)]^2}{[\underline{\phi}^T(i-d)\underline{\phi}(i-d) + c]} \right\} < \infty \quad (4.2.1.7)$$

which also implies:

$$\lim_{k \rightarrow \infty} \sum_{i=1}^k \left\{ \frac{[\tilde{\hat{\Theta}}^T(i-1)\underline{\phi}(i-d)]^2 [\underline{\phi}^T(i-d)\underline{\phi}(i-d) + c]}{[\underline{\phi}^T(i-d)\underline{\phi}(i-d) + c]^2} \right\} < \infty \quad (4.2.1.8)$$

Using (4.2.1.1) in (4.2.1.8):

$$\lim_{k \rightarrow \infty} \sum_{i=1}^k \|\hat{\Theta}(i) - \hat{\Theta}(i-1)\|^2 < \infty \quad (4.2.1.9)$$

which implies:

$$\lim_{k \rightarrow \infty} \|\hat{\Theta}(k) - \hat{\Theta}(k-1)\| = 0 \quad (4.2.1.10)$$

By using the triangle rule (Schwartz's inequality):

$$\begin{aligned} \|\hat{\Theta}(i) - \hat{\Theta}(i-d)\|^2 &= \|\hat{\Theta}(i) - \hat{\Theta}(i-1) + \hat{\Theta}(i-1) - \hat{\Theta}(i-2) \dots \\ &\quad + \hat{\Theta}(i-d+1) - \hat{\Theta}(i-d)\|^2 \leq d \{ \|\hat{\Theta}(i) - \hat{\Theta}(i-1)\|^2 + \dots \\ &\quad + \|\hat{\Theta}(i-d+1) - \hat{\Theta}(i-d)\|^2 \} \end{aligned} \quad (4.2.1.11)$$

Using (4.2.1.10) and (4.2.1.11) with a finite delay d :

$$\lim_{k \rightarrow \infty} \sum_{i=d}^k \|\hat{\Theta}(i) - \hat{\Theta}(i-d)\|^2 < \infty \quad (4.2.1.12)$$

from which:

$$\lim_{k \rightarrow \infty} \|\hat{\underline{\Theta}}(k) - \hat{\underline{\Theta}}(k-d)\| = 0$$

(4.2.1.13)

follows directly. Finally, combining (4.2.1.7) and (4.2.1.13), it can be stated that the “normalized prediction error” converges to zero:

$$\lim_{k \rightarrow \infty} \frac{[\tilde{\underline{\Theta}}^T(k-d)\underline{\phi}(k-d)]^2}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} = 0$$

(4.2.1.14)

which is the desired result. The “normalization” here again refers to the presence of $\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c$ in the denominator of (4.2.1.14).

The normalized gradient algorithm can be applied without modification to a system with time-variant parameters, although as mentioned earlier the convergence analysis is generally not valid. Perhaps the major disadvantage of the method is its slow convergence rate when compared to most least squares algorithms.

4.2.2 Least Squares Parameter Estimation

4.2.2.1 Normal Least Squares

The normal least squares parameter estimation algorithm applied to the system of (4.1.1.8) and estimated system (4.1.1.9) is:

$$\hat{\underline{\Theta}}(k) = \hat{\underline{\Theta}}(k-1) + \frac{\underline{P}(k-d-1)\underline{\phi}(k-d)[\underline{e}^T \underline{x}(k) - \underline{e}'^T \underline{A}' \underline{x}(k-d) - \hat{\underline{\Theta}}^T(k-1)\underline{\phi}(k-d)]}{1 + \underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)}$$

(4.2.2.1.1)

$$\underline{P}(k-d) = \underline{P}(k-d-1) - \frac{\underline{P}(k-d-1)\underline{\phi}(k-d)\underline{\phi}^T(k-d)\underline{P}(k-d-1)}{1 + \underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)}$$

(4.2.2.1.2)

$$\underline{P}(-1) = \epsilon \underline{I}$$

(4.2.2.1.3)

where $k \geq d$ and $0 < \epsilon < \infty$.

To show that the parameter estimates are bounded, use the Liapunov function candidate:

$$V(k) = \tilde{\underline{\Theta}}^T(k) \underline{P}^{-1}(k-d) \tilde{\underline{\Theta}}(k)$$

(4.2.2.1.4)

Using the definition of $\tilde{\underline{\Theta}}(k)$ and (4.2.2.1.1):

$$\tilde{\underline{\Theta}}(k) = \tilde{\underline{\Theta}}(k-1) - \frac{\underline{P}(k-d-1) \underline{\phi}(k-d) [\underline{e}^T \underline{x}(k) - \underline{e}^T \underline{A}' \underline{x}(k-d) - \hat{\underline{\Theta}}^T(k-1) \underline{\phi}(k-d)]}{1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)}$$

(4.2.2.1.5)

Calculate the first backward difference of the Liapunov function candidate (4.2.2.1.4):

$$\Delta V(k) = V(k) - V(k-1) = \tilde{\underline{\Theta}}^T(k) \underline{P}^{-1}(k-d) \tilde{\underline{\Theta}}(k) - \tilde{\underline{\Theta}}^T(k-1) \underline{P}^{-1}(k-d-1) \tilde{\underline{\Theta}}(k-1)$$

(4.2.2.1.6)

Using (4.2.2.1.5) and (4.1.1.8):

$$\tilde{\underline{\Theta}}(k) = \left\{ \underline{I} - \frac{\underline{P}(k-d-1) \underline{\phi}(k-d) \underline{\phi}^T(k-d)}{1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)} \right\} \tilde{\underline{\Theta}}(k-1)$$

(4.2.2.1.7)

Post-multiplying the expression in brackets by $\underline{P}(k-d-1)$ and pre-multiplying the term outside the brackets by $\underline{P}^{-1}(k-d-1)$:

$$\tilde{\underline{\Theta}}(k) = \left\{ \underline{P}(k-d-1) - \frac{\underline{P}(k-d-1) \underline{\phi}(k-d) \underline{\phi}^T(k-d) \underline{P}(k-d-1)}{1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)} \right\} \underline{P}^{-1}(k-d-1) \tilde{\underline{\Theta}}(k-1)$$

(4.2.2.1.8)

Substituting (4.2.2.1.2) into (4.2.2.1.8):

$$\tilde{\underline{\mathbf{Q}}}(k) = \underline{\mathbf{P}}(k-d)\underline{\mathbf{P}}^{-1}(k-d-1)\tilde{\underline{\mathbf{Q}}}(k-1) \quad (4.2.2.1.9)$$

Substituting (4.2.2.1.9) into (4.2.2.1.6) gives:

$$\Delta V(k) = [\tilde{\underline{\mathbf{Q}}}(k) - \tilde{\underline{\mathbf{Q}}}(k-1)]^T \underline{\mathbf{P}}^{-1}(k-d-1)\tilde{\underline{\mathbf{Q}}}(k-1) \quad (4.2.2.1.10)$$

Again using (4.2.2.1.5):

$$\Delta V(k) = -\frac{[\tilde{\underline{\mathbf{Q}}}^T(k-1)\underline{\phi}(k-d)]^2}{1 + \underline{\phi}^T(k-d)\underline{\mathbf{P}}(k-d-1)\underline{\phi}(k-d)} \quad (4.2.2.1.11)$$

which is non-positive.

To show that $\tilde{\underline{\mathbf{Q}}}(k) = \underline{\mathbf{0}}$ is a unique minimum and that the Liapunov function candidate is always non-negative, it is necessary to show that the minimum eigenvalue of $\underline{\mathbf{P}}^{-1}(k-d)$ is > 0 . It is useful to apply the matrix inversion lemma to (4.2.2.1.2):

$$\underline{\mathbf{P}}^{-1}(k) = \underline{\mathbf{P}}^{-1}(k-1) + \underline{\phi}(k)\underline{\phi}^T(k) \quad (4.2.2.1.12)$$

Using this result and the fact that $\underline{\mathbf{P}}(-1)$ is positive definite (from 4.2.2.1.3) shows that $\underline{\mathbf{P}}^{-1}(k)$ remains positive definite, as the addition of the positive semidefinite matrix $\underline{\phi}(k)\underline{\phi}^T(k)$ will not affect the positive definiteness of $\underline{\mathbf{P}}^{-1}(k)$ for any k . Thus, the minimum eigenvalue of $\underline{\mathbf{P}}^{-1}(k)$ is always > 0 which confirms that $V(k)$ is a Liapunov function of the estimator system (4.2.2.1.1), (4.2.2.1.2), (4.2.2.1.3) and therefore that the parameter estimates will remain bounded.

To show that the normalized prediction error converges to zero, evaluate $V(k)$ from $k = 0$ to $k = k$ using (4.2.2.1.11):

$$V(k) = V(k=0) - \sum_{i=1}^k \left\{ \frac{[\tilde{\Theta}^T(k-1)\underline{\phi}(k-d)]^2}{1 + \underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)} \right\} \quad (4.2.2.1.13)$$

Using the fact that $V(k)$ is non-negative and bounded implies that:

$$\lim_{k \rightarrow \infty} \sum_{i=1}^k \left\{ \frac{[\tilde{\Theta}^T(k-1)\underline{\phi}(k-d)]^2}{1 + \underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)} \right\} < \infty \quad (4.2.2.1.14)$$

and that:

$$\lim_{k \rightarrow \infty} \sum_{i=1}^k \left\{ \frac{[\tilde{\Theta}^T(k-1)\underline{\phi}(k-d)]^2 [1 + \underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)]}{[1 + \underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)]^2} \right\} < \infty \quad (4.2.2.1.15)$$

Using (4.2.2.1.14) and (4.2.2.1.15):

$$\lim_{k \rightarrow \infty} \sum_{i=1}^k \left\{ \frac{[\tilde{\Theta}^T(k-1)\underline{\phi}(k-d)]^2 \underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)}{[1 + \underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)]^2} \right\} < \infty \quad (4.2.2.1.16)$$

Note that (4.2.2.1.16) can be expressed as:

$$\lim_{k \rightarrow \infty} \sum_{i=1}^k \left\{ \frac{[\tilde{\Theta}^T(k-1)\underline{\phi}(k-d)]^2 \underline{\phi}^T(k-d)\underline{P}^2(k-d-1)\underline{P}^{-1}(k-d-1)\underline{\phi}(k-d)}{[1 + \underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)]^2} \right\} < \infty \quad (4.2.2.1.17)$$

As shown earlier, $\underline{P}^{-1}(k)$ is positive definite for all k so that (4.2.2.1.17) implies:

$$\lim_{k \rightarrow \infty} \sum_{i=1}^k \left\{ \frac{[\tilde{\Theta}^T(k-1)\underline{\phi}(k-d)]^2 \underline{\phi}^T(k-d)\underline{P}^2(k-d-1)\underline{\phi}(k-d)}{[1 + \underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)]^2} \right\} < \infty \quad (4.2.2.1.18)$$

Using (4.2.2.1.1) this is also:

$$\lim_{k \rightarrow \infty} \sum_{i=1}^k \|\hat{\underline{\theta}}(i) - \hat{\underline{\theta}}(i-1)\|^2 < \infty \quad (4.2.2.1.19)$$

As in Section 4.2.1, this can be shown to lead to:

$$\lim_{k \rightarrow \infty} \|\hat{\underline{\theta}}(k) - \hat{\underline{\theta}}(k-d)\|^2 = 0 \quad (4.2.2.1.20)$$

Together with (4.2.2.1.14), it can finally be stated that:

$$\lim_{k \rightarrow \infty} \frac{[\tilde{\underline{\theta}}^T(k-d)\underline{\phi}(k-d)]^2}{1 + \underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)} = 0 \quad (4.2.2.1.21)$$

which is the desired result.

The normal least squares algorithm generally exhibits faster convergence than the normalized gradient algorithm but is not suited directly to handling time-variant parameters. Several versions of the algorithm are available (see Sections 4.2.2.3 and 4.2.2.4) which are better suited to the time-variant parameter case.

4.2.2.2 Weighted Least Squares

The weighted least squares algorithm allows for the selective weighting of data based on its expected information content. It has the form:

$$\hat{\underline{\theta}}(k) = \hat{\underline{\theta}}(k-1) + \frac{a(k)\underline{P}(k-d-1)\underline{\phi}(k-d)[\underline{e}^T \underline{x}(k) - \underline{e}^{T'} \underline{A}' \underline{x}(k-d) - \hat{\underline{\theta}}^T(k-1)\underline{\phi}(k-d)]}{1 + a(k)\underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)} \quad (4.2.2.2.1)$$

$$\underline{P}(k-d) = \underline{P}(k-d-1) - \frac{a(k)\underline{P}(k-d-1)\underline{\phi}(k-d)\underline{\phi}^T(k-d)\underline{P}(k-d-1)}{1 + a(k)\underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)} \quad (4.2.2.2.2)$$

$$\underline{P}(-1) = \epsilon \underline{I} \quad (4.2.2.2.3)$$

where $k \geq d$, $0 < \epsilon < \infty$, and $a(k)$ non-negative. The analysis of Section 4.2.2.1 carries over directly to this algorithm as well, and is therefore omitted for brevity. As $a(k) \rightarrow 0$, the weighted least squares converges to the "orthogonal projection" algorithm (see [72] which is simply a recursive solution to a set of n simultaneous linear equations. In general, however, susceptibility to noise of the algorithm increases as $a(k)$ decreases which limits the use of this extreme case of $a(k)$ in practice.

4.2.2.3 Forgetting Factor Least Squares

The forgetting factor least squares algorithm is so called because it discounts old data at a certain rate in favor of new data. It has the form:

$$\begin{aligned} \hat{\underline{\epsilon}}(k) &= \hat{\underline{\epsilon}}(k-1) + \\ &+ \frac{\underline{P}(k-d-1)\underline{\phi}(k-d)[\underline{\epsilon}^T \underline{x}(k) - \underline{\epsilon}^T \underline{A}' \underline{x}(k-d) - \hat{\underline{\epsilon}}^T(k-1)\underline{\phi}(k-d)]}{a(k) + \underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)} \end{aligned} \quad (4.2.2.3.1)$$

$$\underline{P}(k-d) = \frac{1}{a(k)} \left\{ \underline{P}(k-d-1) - \frac{\underline{P}(k-d-1)\underline{\phi}(k-d)\underline{\phi}^T(k-d)\underline{P}(k-d-1)}{a(k) + \underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)} \right\} \quad (4.2.2.3.2)$$

$$\underline{P}(-1) = \epsilon \underline{I}$$

(4.2.2.3.3)

where $k \geq d$, $0 < \epsilon < \infty$, and $0 < a(k) < 1$.

As in Section 4.2.2.1, it is necessary to show that the parameter estimates are bounded. Again use the Liapunov function candidate:

$$V(k) = \tilde{\underline{\Theta}}^T(k) \underline{P}^{-1}(k-d) \tilde{\underline{\Theta}}(k)$$

(4.2.2.3.4)

Using the definition of $\tilde{\underline{\Theta}}(k)$ and (4.2.2.3.1):

$$\tilde{\underline{\Theta}}(k) = \tilde{\underline{\Theta}}(k-1) - \frac{\underline{P}(k-d-1) \underline{\phi}(k-d) [\underline{e}^T \underline{x}(k) - \underline{e}'^T \underline{A}' \underline{x}(k-d) - \hat{\underline{\Theta}}^T(k-1) \underline{\phi}(k-d)]}{a(k) + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)}$$

(4.2.2.3.5)

Calculate the first backward difference of the Liapunov function candidate (4.2.2.3.4):

$$\Delta V(k) = V(k) - V(k-1) = \tilde{\underline{\Theta}}^T(k) \underline{P}^{-1}(k-d) \tilde{\underline{\Theta}}(k) - \tilde{\underline{\Theta}}^T(k-1) \underline{P}^{-1}(k-d-1) \tilde{\underline{\Theta}}(k-1)$$

(4.2.2.3.6)

Following the method of Section 4.2.2.1:

$$\begin{aligned} \Delta V(k) = & - \frac{[\tilde{\underline{\Theta}}^T(k-1) \underline{\phi}(k-d)]^2}{a(k) + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)} + \\ & + [a(k) - 1] \tilde{\underline{\Theta}}^T(k-1) \underline{P}^{-1}(k-d-1) \tilde{\underline{\Theta}}(k-1) \end{aligned}$$

(4.2.2.3.7)

Applying the matrix inversion lemma to (4.2.2.3.2) gives:

$$\underline{P}^{-1}(k) = a(k)\underline{P}^{-1}(k-1) + \underline{\phi}(k)\underline{\phi}^T(k) \quad (4.2.2.3.8)$$

which can again be shown to keep $\underline{P}^{-1}(k)$ positive definite for all k for $0 < a(k) < 1$. Consequently, $V(k)$ is non-negative. Using this fact together with $0 < a(k) < 1$ shows that (4.2.2.3.7) is non-positive and hence $V(k)$ is a Liapunov function of the estimator system (4.1.1.8) and (4.2.2.3.1)-(4.2.2.3.2). The analogous steps of Section 4.2.2.1 can be used to show that the normalized prediction error converges to zero for this algorithm as well.

It is interesting to note that (4.2.2.3.7) can be re-written as:

$$V(k) = a(k)V(k-1) - \frac{[\tilde{\mathbf{Q}}^T(k-1)\underline{\phi}(k-d)]^2}{a(k) + \underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d-1)} \quad (4.2.2.3.9)$$

which provides exponential convergence of the Liapunov function, but does not imply the same about the parameter errors themselves unless conditions on the elements of the regressor vector are imposed. Although this algorithm is well suited for time-variant parameters since it discards old data, the matrix $\underline{P}(k)$ can grow without bound if the regressor vector does not satisfy a “sufficient richness” condition (to be presented in Section 4.2.3). A possible solution to this difficulty is to monitor the “size” of $\underline{P}(k)$ (e.g. its maximum eigenvalue or trace) and shift to a modified version (e.g. normal or weighted least squares) of the algorithm when a pre-specified bound has been reached.

4.2.2.4 Covariance Resetting Least Squares

The covariance resetting least squares algorithm attempts to maintain the high initial convergence of the normal least squares algorithm by resetting the “covariance matrix” $\underline{P}(k)$ (so named from its Kalman filtering interpretation) at periodic intervals.

It has the form:

$$\hat{\underline{\mathbf{e}}}(k) = \hat{\underline{\mathbf{e}}}(k-1) + \frac{\underline{\mathbf{P}}(k-d-1)\underline{\phi}(k-d)[\underline{\mathbf{e}}^T \underline{\mathbf{x}}(k) - \underline{\mathbf{e}}'^T \underline{\mathbf{A}}' \underline{\mathbf{x}}(k-d) - \hat{\underline{\mathbf{e}}}^T(k-1)\underline{\phi}(k-d)]}{1 + \underline{\phi}^T(k-d)\underline{\mathbf{P}}(k-d-1)\underline{\phi}(k-d)} \quad (4.2.2.4.1)$$

$$\underline{\mathbf{P}}(-1) = \epsilon \underline{\mathbf{I}} \quad (4.2.2.4.2)$$

At the times when resetting occurs:

$$\underline{\mathbf{P}}(k-1) = \alpha (k-1) \underline{\mathbf{I}} \quad (4.2.2.4.3)$$

At other times, the normal least squares update for $\underline{\mathbf{P}}(k)$ is used, i.e.:

$$\underline{\mathbf{P}}(k-d) = \underline{\mathbf{P}}(k-d-1) - \frac{\underline{\mathbf{P}}(k-d-1)\underline{\phi}(k-d)\underline{\phi}^T(k-d)\underline{\mathbf{P}}(k-d-1)}{1 + \underline{\phi}^T(k-d)\underline{\mathbf{P}}(k-d-1)\underline{\phi}(k-d)} \quad (4.2.2.4.4)$$

where $k \geq d, 0 < \epsilon < \infty$, and $0 < \alpha (k-1) < \infty$. Proofs to show parameter boundedness and prediction error convergence are given in [72].

The algorithm is also suited for use when the parameters are time-variant. In this case some auxiliary signal change (such as an increase in prediction error) indicates a significant parameter change has occurred, and causes the covariance to be reset to enhance convergence to the new parameter value. As pointed out earlier, however, the boundedness and convergence proofs derived in the time-invariant parameter case do not hold unless detailed knowledge is available on the time variation of the actual parameters.

4.2.3 Parameter Convergence

The parameter estimators of Sections 4.2.1 and 4.2.2 have been shown to result in bounded parameter estimates and convergence of the normalized prediction error to zero. Without imposing special conditions on the regressor vector, however, these results do not also necessarily imply that the parameter estimates converge to the true parameter values. In the pure parameter estimation problem, it is therefore desirable to have some control over the regressor vector in order that these special conditions are met. In the adaptive control context, however, the actual convergence of the parameter values to their true values usually takes on a lesser importance. This is because the goal of the control problem is commonly to have some performance criteria minimized (such as tracking error) rather than the explicit minimization of parameter estimation error. Nonetheless, a faster parameter convergence rate will generally result in the controller being better able to meet its performance objectives. Further, if one of the goals of the controller is to also provide information on the true parameter values as an auxiliary function (such as system identification), then these special conditions on the regressor vector are of interest. More will be said on the matter of parameter convergence during adaptive control in a later section.

Conditions on the regressor vector can be expressed both in the frequency and in the time domain. Because the interest here is on real time estimation and control, the discussion will be limited to conditions expressed in the time domain.

An analysis of parameters convergence properties when using the normalized gradient algorithm is given in [72]. The algorithm is found to be asymptotically convergent when the following “sufficient excitation” condition is satisfied:

$$\lim_{k \rightarrow \infty} \lambda_{\min} \left\{ \sum_{i=0}^k \underline{\phi}(i) \underline{\phi}^T(i) \right\} = \infty \quad (4.2.3.1)$$

where $\lambda_{\min}\{\cdot\}$ is the minimum eigenvalue. (4.2.3.1) can also be expressed as:

$$\lim_{k \rightarrow \infty} \sum_{i=0}^k \underline{\phi}(i) \underline{\phi}^T(i) = \infty \quad (4.2.3.2)$$

Furthermore, the algorithm is exponentially convergent when the following "persistent excitation" condition is satisfied:

$$\lambda_{\min} \left\{ \sum_{i=k}^{k+\ell} \underline{\phi}(i) \underline{\phi}^T(i) \right\} > 0 \quad (4.2.3.3)$$

for all k , where ℓ is an integer $\geq n$. This can also be expressed as:

$$p_1 I > \sum_{i=k}^{i+\ell} \underline{\phi}(i) \underline{\phi}^T(i) > p_2 I \quad (4.2.3.4)$$

for all k , where $p_1, p_2 > 0$.

The persistent excitation condition of (4.2.3.3) is seen to be more stringent than the sufficient excitation condition of (4.2.3.2). This is because (4.2.3.3) requires that a lower bound away from zero exist on the matrix $\underline{\phi}(i) \underline{\phi}^T(i)$ during a fixed interval, and therefore "persistently." (4.2.3.2) only requires that a lower bound away from zero exists for a subset of time - i.e., the matrix $\underline{\phi}(i) \underline{\phi}^T(i)$ may remain singular for an indefinite period.

Analysis of parameter convergence using the normal least squares algorithm is done by recalling from Section 4.2.2.1 that the expression:

$$V(k) = \tilde{\underline{\Theta}}^T(k) \underline{P}^{-1}(k-d) \tilde{\underline{\Theta}}(k) \quad (4.2.3.5)$$

was shown to be a Liapunov function, which implies it is a non-negative and non-increasing function. Therefore:

$$V(k) \geq \lambda_{\min} \{ \underline{P}^{-1}(k-d) \} \tilde{\underline{\Theta}}^T(k) \tilde{\underline{\Theta}}(k) \quad (4.2.3.6)$$

By summing the expression for $\underline{P}^{-1}(i)$ from $i = 0$ to $i = k - d$ using (4.2.2.1.12):

$$\underline{P}^{-1}(k-d) = \underline{P}^{-1}(-1) + \sum_{i=0}^{k-d} \underline{\phi}(i) \underline{\phi}^T(i) \quad (4.2.3.7)$$

From (4.2.3.6), it is clear that if:

$$\lim_{k \rightarrow \infty} \lambda_{\min} \{ \underline{P}^{-1}(k-d) \} = \infty \quad (4.2.3.8)$$

then for $V(k)$ to remain bounded:

$$\lim_{k \rightarrow \infty} \|\tilde{\underline{\Theta}}(k)\|^2 = 0 \quad (4.2.3.9)$$

The condition of (4.2.3.8) can be related to a condition on $\underline{\phi}(k)$ using (4.2.3.7):

$$\lim_{k \rightarrow \infty} \lambda_{\min} \left\{ \sum_{i=0}^{k-d} \underline{\phi}(i) \underline{\phi}^T(i) \right\} = \infty \quad (4.2.3.10)$$

since $\underline{P}^{-1}(-1)$ is bounded. Condition (4.2.3.10) is thus equivalent to the sufficient excitation condition of (4.2.3.1) and guarantees that the least squares algorithms and its variants are asymptotically convergent.

Under the persistent excitation condition of (4.2.3.3), it is possible to show that the forgetting factor least squares variant produces parameter estimates that converge

exponentially to their actual values. Recall from Section 4.2.2.3 that the Liapunov function was exponentially convergent. To prove that the parameter estimation error $\tilde{\Theta}(k)$ is also exponentially convergent, it is necessary to show that $\underline{P}^{-1}(k-d)$ used in the Liapunov function is lower bounded. Using the matrix inversion lemma with (4.2.2.3.2), the propagation of $\underline{P}^{-1}(k-d)$ is:

$$\underline{P}^{-1}(k-d) = a(k)\underline{P}^{-1}(k-d-1) + \underline{\phi}(k-d)\underline{\phi}^T(k-d) \quad (4.2.3.11)$$

Summing both sides from $k = k$ to $k = \ell$ and using $0 < a(k) < 1$ implies:

$$\underline{P}^{-1}(k-d+\ell) \geq \min_{k=k \rightarrow k=\ell} \{a(k)\} \underline{P}^{-1}(k-d-1) + \sum_{i=k-d}^{k-d+\ell} \underline{\phi}(i)\underline{\phi}^T(i) \quad (4.2.3.12)$$

Using (4.2.3.3), the right side of (4.2.3.12) is lower bounded which implies that the left side is also. In this case, the exponential convergence of the Liapunov function also implies the exponential convergence of the parameter estimates.

4.3 Constrained Parameter Estimation

There are often known bounds on the parameters being estimated which can guide the parameter estimation process. These bounds typically consist of a known sign on the control input or the knowledge that the system being estimated is stable. Methods for properly utilizing these known bounds are presented in [72], and therefore only a brief summary will be presented here for completeness. The assumption is that a closed convex region exists in parameter space which includes a point representing the true parameter values.

4.3.1 Normalized Gradient

If the algorithm leads to a $\hat{\underline{Q}}(k)$ which is outside the constraining region, this estimate should be projected orthogonally onto the surface of the region before continuing. For a convex region, this ensures that the resulting (modified) estimate is “closer” to the real parameter value point than the original estimate. Thus, the $V(k)$ of (4.2.1.2) remains a Liapunov function as before. This orthogonal projection becomes trivial if the constraining region is a hypercube in parameter space, as the parameter estimate simply is changed to the lower or upper limit of the parameter, as appropriate, if the estimate is outside of these bounds.

4.3.2 Least Squares

The parameter error vectors are weighted by $\underline{P}^{-1}(k)$ in the Liapunov function (4.2.2.1.4) for the least squares convergence analysis and thus requires a different procedure when applying the parameter constraints.

If $\hat{\underline{Q}}(k)$ is outside the constrained region, then the following steps are required:

1. Transform the coordinate basis for the parameter space by defining:

$$\underline{\rho} = \underline{P}^{-1/2}(k-d)\underline{\Theta} \quad (4.3.2.1)$$

where:

$$\underline{P}^{-1}(k-d) = \underline{P}^{-T/2}(k-d)\underline{P}^{-1/2}(k-d) \quad (4.3.2.2)$$

2. Transform $\hat{\underline{\Theta}}(k)$ into the new coordinate system using (4.3.2.1):

$$\hat{\underline{\rho}}(k) = \underline{P}^{-1/2}(k-d)\hat{\underline{\Theta}}(k) \quad (4.3.2.3)$$

3. Orthogonally project $\hat{\underline{\rho}}(k)$ onto the transformation of the constrained region and let this point be $\underline{\hat{\rho}}'(k)$.
4. Transform $\underline{\hat{\rho}}'(k)$ back into the original coordinate system using the inverse of the transformation (4.3.2.1):

$$\hat{\underline{\Theta}}(k) = \underline{P}^{1/2}(k-d)\underline{\hat{\rho}}'(k) \quad (4.3.2.4)$$

The motivation for using transformation (4.3.2.1) can be seen by examining the Liapunov function of (4.2.2.1.4):

$$V(k) = \tilde{\underline{\Theta}}^T(k)\underline{P}^{-1}(k-d)\tilde{\underline{\Theta}}(k) \quad (4.3.2.5)$$

Introducing the transformation (4.3.2.1) into (4.3.2.5):

$$V(k) = \tilde{\underline{\rho}}^T(k)\tilde{\underline{\rho}}(k) \quad (4.3.2.6)$$

where $\tilde{\underline{\rho}}(k) = \underline{P}(k-d)^{-1/2}\underline{\Theta} - \hat{\underline{\rho}}(k)$. By orthogonally projecting $\hat{\underline{\rho}}(k)$ on to the transformed constraint region to get $\hat{\underline{\rho}}'(k)$, the resulting (modified) estimate is closer to the real (transformed) parameter values and thus (4.3.2.6) and therefore (4.3.2.5) remains a Liapunov function. To add to the discussion in [72], this procedure becomes straightforward if the constraining region is a hypercube in parameter space. Since (4.3.2.1) is a linear transformation, the hypercube in the original parameter space transforms into a hypercube in the transformed parameter space as well. The steps to apply the constraints in this case are as follows:

1. Determine in the original parameter space which parameters of the n dimensional parameter vector exceed their upper or lower bounds.
2. Pick one of the affected parameters which occupies the i^{th} position in this parameter vector and designate this parameter Θ_i . Form the vector \underline{g} which is equal in length to $\underline{\Theta}$ and in which all elements are zero except for the i^{th} element which is set equal to one, i.e.:

$$\underline{g}_i = [0 \cdots 0 \quad 1 \quad 0 \cdots 0] \quad (4.3.2.7)$$

3. Let the parameter constraint boundary be given by:

$$\underline{g}_i^T \underline{\Theta} = q_i \quad (4.3.2.8)$$

where q_i is the upper or lower limit which has been exceeded, as appropriate, of parameter i .

4. In the transformed parameter space, this becomes, using (4.3.2.1):

$$\underline{g}_i^T \underline{P}^{1/2}(k-d)\underline{\rho} = q_i \quad (4.3.2.9)$$

5. Obtain $\hat{\rho}(k)$ using (4.3.2.3) and project this onto the transformed constraint boundary given by (4.3.2.9) using the orthogonalized projection formula:

$$\hat{\rho}'(j) = \hat{\rho}'(j-1) + \frac{\underline{Q}(j-2)\underline{P}^{T/2}(k-d)\underline{g}_i[q_i - \underline{g}_i^T \underline{P}^{1/2}(k-d)\hat{\rho}(j-1)]}{\underline{g}_i^T \underline{P}^{T/2}(k-d)\underline{Q}(j-2)\underline{P}^{1/2}(k-d)\underline{g}_i} \quad (4.3.2.10)$$

$$\underline{Q}(j-1) = \underline{Q}(j-2) - \frac{\underline{Q}(j-2)\underline{P}^{T/2}(k-d)\underline{g}_i \underline{g}_i^T \underline{P}^{1/2}(k-d)\underline{Q}(j-2)}{\underline{g}_i^T \underline{P}^{T/2}(k-d)\underline{Q}(j-2)\underline{P}^{1/2}(k-d)\underline{g}_i} \quad (4.3.2.11)$$

$$\underline{Q}(0) = \underline{I} \quad (4.3.2.12)$$

$$\hat{\rho}'(1) = \hat{\rho}(k) \quad (4.3.2.13)$$

where $j \geq 2$.

6. After each iteration of this algorithm, the result $\hat{\rho}'(j)$ is transformed back to the original parameter space to see if the resulting estimate has returned to the constrained region, i.e., using (4.3.2.4):

$$\hat{\hat{\rho}}'(j) = \hat{\hat{\rho}}'(j-1) + \frac{\underline{P}^{1/2}(k-d)\underline{Q}(j-2)\underline{P}^{T/2}(k-d)\underline{g}_i[q_i - \hat{\hat{\rho}}'_i(j-1)]}{\underline{g}_i^T \underline{P}^{T/2}(k-d)\underline{Q}(j-2)\underline{P}^{1/2}(k-d)\underline{g}_i} \quad (4.3.2.14)$$

7. If $\hat{\hat{\rho}}'(j)$ reaches the constrained region, the iteration is stopped and $\hat{\hat{\rho}}(k)$ set equal to $\hat{\hat{\rho}}'(j)$.

4.4 Parameter Estimation with Bounded Disturbances

The parameter estimation schemes of Section 4.2.1 and 4.2.2 can be extended to systems with bounded disturbances. That is, the system description becomes (4.1.1.8) where $|d(k)| \leq D(k)$. This is accomplished by passing the estimation error through a “dead zone” function such that parameter estimation does not occur unless the estimation error is larger than that which could be attributed to the disturbance alone. This is implemented in a slightly different manner for the normalized gradient and the least squares algorithms.

4.4.1 Normalized Gradient

The system description of (4.1.1.8) is used:

$$\underline{e}^T \underline{x}(k+d) = \underline{e}^T \underline{A}' \underline{x}(k) + \underline{\Theta}^T \underline{\phi}(k) + d(k) \quad (4.4.1.1)$$

where $|d(k)| \leq D(k) < \infty$. As in Section 4.2, it is necessary to show that the parameter estimates are bounded and that the normalized prediction error converges to some limit. The estimated system corresponding to (4.4.1.1) is, from (4.1.1.9):

$$\underline{e}^T \hat{\underline{x}}(k+d) = \underline{e}^T \underline{A}' \underline{x}(k) + \hat{\underline{\Theta}}^T(k) \underline{\phi}(k) \quad (4.4.1.2)$$

with prediction error:

$$\underline{e}^T \hat{\underline{x}}(k+d) = \tilde{\underline{\Theta}}^T(k) \underline{\phi}(k) + d(k) \quad (4.4.1.3)$$

Because of the stochastic nature of $d(k)$, it is obvious that the prediction error cannot in general be reduced to zero, but rather can be shown to approach a lower limit based on the size of $D(k)$ which bounds $d(k)$.

The modified algorithm is given by:

$$\hat{\underline{\Theta}}(k) = \hat{\underline{\Theta}}(k-1) + \frac{a(k) \underline{\phi}(k-d) g_{\Delta}(k)}{\underline{\phi}^T(k-d) \underline{\phi}(k-d) + c} \quad (4.4.1.4)$$

where $k \geq d, 0 < a(k) < 2, c > 0$, and:

$$g_{\Delta}(k) = \begin{cases} g(k) - R(k) \text{sgn}\{g(k)\} & \text{if } |g(k)| \geq R(k) \\ 0 & \text{otherwise} \end{cases} \quad (4.4.1.5)$$

where $R(k) \geq D(k)$, and:

$$g(k) = \underline{e}^T \underline{x}(k) - \underline{e}^T \underline{A}' \underline{x}(k-d) - \hat{\underline{\Theta}}^T(k-1) \underline{\phi}(k-d) \quad (4.4.1.6)$$

To show the estimates are bounded, use the Liapunov function candidate:

$$V(k) = \|\tilde{\underline{\Theta}}(k)\|^2 \quad (4.4.1.7)$$

Using (4.4.1.4) and the definition of $\tilde{\underline{\Theta}}(k)$:

$$\tilde{\underline{\mathbf{Q}}}(k) = \tilde{\underline{\mathbf{Q}}}(k-1) - \frac{a(k)\underline{\phi}(k-d)g_{\Delta}(k)}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} \quad (4.4.1.8)$$

Calculate the first backward difference of the Liapunov function candidate (4.4.1.7):

$$\Delta V(k) = V(k) - V(k-1) = \|\tilde{\underline{\mathbf{Q}}}(k)\|^2 - \|\tilde{\underline{\mathbf{Q}}}(k-1)\|^2 \quad (4.4.1.9)$$

Substitution of (4.4.1.8) into (4.4.1.9) gives:

$$\Delta V(k) = \frac{-2a(k)[g(k) - d(k-d)]g_{\Delta}(k)}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} + \frac{a^2(k)\underline{\phi}^T(k-d)\underline{\phi}(k-d)g_{\Delta}^2(k)}{[\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c]^2} \quad (4.4.1.10)$$

Therefore:

$$\Delta V(k) \leq \frac{-2a(k)[g(k) - d(k-d)]g_{\Delta}(k)}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} + \frac{a^2(k)g_{\Delta}^2(k)}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} \quad (4.4.1.11)$$

This can be rewritten using (4.4.1.6) when $|g(k)| \geq R(k)$:

$$\Delta V(k) \leq \frac{-2a(k)[g_{\Delta}(k) + R(k)\text{sgn}\{g(k)\} - d(k-d)]g_{\Delta}(k) + a^2(k)g_{\Delta}^2(k)}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} \quad (4.4.1.12)$$

Given that $R(k) \geq D(k-d) \geq d(k-d)$ and that $\text{sgn}\{g(k)\} = \text{sgn}\{g_{\Delta}(k)\}$, $\Delta V(k)$ of (4.4.1.12) is non-positive. Finally, since the $V(k)$ of (4.4.1.7) is non-negative and has a unique minimum at $\tilde{\underline{\mathbf{Q}}}(k) = \underline{\mathbf{0}}$, $V(k)$ is a Liapunov function of the modified estimator system with bounded disturbance (4.4.1.1), and (4.4.1.4) - (4.4.1.6). Note that when $|g(k)| < R(k)$, parameter estimator does not occur and thus $\Delta V(k) = 0$, which is again non-positive.

To show that the size of the normalized prediction error converges to $R(k)$ (which, given the disturbance environment, is the best that can be expected), evaluate $\|\tilde{\Theta}(k)\|^2$ from $k = 0$ to $k = k$ using (4.4.1.12):

$$\begin{aligned} \|\tilde{\Theta}(k)\|^2 &\leq \|\tilde{\Theta}(k=0)\|^2 + \sum_{i=1}^k \left\{ \frac{a(k)}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} \right\} \\ &\cdot \left\{ [a(k) - 2]g_{\Delta}^2(k) - 2g_{\Delta}(k)[R(k)\text{sgn}\{g(k)\} - d(k-d)] \right\} \end{aligned} \quad (4.4.1.13)$$

Since $\|\tilde{\Theta}(k)\|^2$ is non-negative and bounded, and using the fact that $\text{sgn}\{g(k)\} = \text{sgn}\{g_{\Delta}(k)\}$ and $g_{\Delta}(k)\text{sgn}\{g_{\Delta}(k)\} = |g_{\Delta}(k)|$:

$$\lim_{k \rightarrow \infty} \frac{[a(k) - 2]g_{\Delta}^2(k) - 2|g_{\Delta}(k)|R(k) + 2g_{\Delta}(k)d(k-d)}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} = 0 \quad (4.4.1.14)$$

which implies, since $R(k) \geq D(k-d) \geq d(k-d)$:

$$\lim_{k \rightarrow \infty} \frac{g_{\Delta}^2(k)}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} = 0 \quad (4.4.1.15)$$

Using (4.4.1.5), this results in:

$$\lim_{k \rightarrow \infty} \frac{[g(k) - R(k)\text{sgn}\{g(k)\}]^2}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} = 0 \quad (4.4.1.16)$$

when $|g(k)| \geq R(k)$. With steps analogous to those of Section 4.2.1, it can be shown that:

$$\lim_{k \rightarrow \infty} \|\hat{\Theta}(k) - \hat{\Theta}(k-d)\| = 0 \quad (4.4.1.17)$$

and (4.4.1.16) can therefore be restated as:

$$\lim_{k \rightarrow \infty} \frac{[\tilde{\underline{\hat{Q}}}^T(k-d)\underline{\phi}(k-d) + d(k-d) - R(k)\text{sgn}\{g(k)\}]^2}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} = 0 \quad (4.4.1.18)$$

which is the desired result, and is valid when $|g(k)| \geq R(K)$.

4.4.2 Least Squares

The modified algorithm for the least squares case is as follows:

$$\hat{\underline{\hat{Q}}}(k) = \hat{\underline{\hat{Q}}}(k-1) + \frac{a(k)\underline{P}(k-d-1)\underline{\phi}(k-d)g(k)}{1 + a(k)\underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)} \quad (4.4.2.1)$$

$$\underline{P}(k-d) = \underline{P}(k-d-1) - \frac{a(k)\underline{P}(k-d-1)\underline{\phi}(k-d)\underline{\phi}^T(k-d)\underline{P}(k-d-1)}{1 + a(k)\underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)} \quad (4.4.2.2)$$

$$\underline{P}(-1) = \epsilon \underline{I} \quad (4.4.2.3)$$

$$a(k) = \begin{cases} 1 & \text{if } g^2(k) > R^2(k)[1 + \underline{\phi}^T(k-d)\underline{P}(k-d-1)\underline{\phi}(k-d)] \\ 0 & \text{otherwise} \end{cases} \quad (4.4.2.4)$$

where $0 < \epsilon < \infty$ and $D(k-d) \leq R(k) < \infty$ with:

$$g(k) = \underline{e}^T \underline{x}(k) - \underline{e}^T \underline{A}' \underline{x}(k-d) - \hat{\underline{\hat{Q}}}^T(k-1)\underline{\phi}(k-d) \quad (4.4.2.5)$$

To show that the estimates are bounded, use the Liapunov function candidate:

$$V(k) = \tilde{\underline{\Theta}}^T(k) \underline{P}^{-1}(k-d) \tilde{\underline{\Theta}}(k) \quad (4.4.2.6)$$

Similar to the steps of Section 4.2.2.1, the first backward difference of the Liapunov function candidate (4.4.2.6) is:

$$\begin{aligned} \Delta V = a(k) & \left\{ \frac{-g^2(k)}{1 + a(k) \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)} + d^2(k-d) + \right. \\ & \left. + g(k)d(k-d)[1 - a(k)] \left[\frac{-a(k) \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)}{1 + a(k) \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)} \right] \right\} \end{aligned} \quad (4.4.2.7)$$

By the selection of $a(k)$ using (4.4.2.4), $\Delta V(k)$ becomes zero if $a(k-1) = 0$, or, if $g^2(k) > R^2(k)[1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)]$:

$$\Delta V(k) = \frac{-g^2(k)}{1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)} + [d(k)]^2 \quad (4.4.2.8)$$

and since $R(k) \geq D(k-d) \geq d(k-d)$, the $\Delta V(k)$ of (4.4.2.8) is non-positive. Finally, since the $V(k)$ of (4.4.2.6) is non-negative and has a unique minimum at $\tilde{\underline{\Theta}}(k) = 0$ (see Section 4.2.2.1), then it is a Liapunov function of the modified estimator system with bounded disturbance (4.4.1.1), and (4.4.2.1)-(4.4.2.4).

To show that the normalized $|g(k)|$ converges to $[1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)]^{1/2} R(k)$ (again, the best to be expected given the disturbance environment), evaluate $V(k)$ from $k = 0$ to $k = k$ using (4.4.2.8):

$$V(k) = V(k=0) - \sum_{i=1}^k \left\{ \frac{g^2(k)}{1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)} - d^2(k-d) \right\} \quad (4.4.2.9)$$

Now since $V(k)$ is non-negative and bounded:

$$\lim_{k \rightarrow \infty} \sum_{i=1}^k \left\{ \frac{g^2(k)}{1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)} - d^2(k-d) \right\} < \infty \quad (4.4.2.10)$$

Since $R(k) \geq D(k-d) \geq d(k-d)$, this implies:

$$\lim_{k \rightarrow \infty} \sum_{i=1}^k \left\{ \frac{g^2(k)}{1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)} - R^2(k) \right\} < \infty \quad (4.4.2.11)$$

which finally implies:

$$\lim_{k \rightarrow \infty} \frac{g^2(k) - [1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)] R^2(k)}{1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)} = 0 \quad (4.4.2.12)$$

which is valid when $|g(k)| \geq [1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)]^{1/2} R(k)$.

With steps analogous to those of Section 4.2.1, it can be shown that:

$$\lim_{k \rightarrow \infty} \|\hat{\Theta}(k) - \hat{\Theta}(k-d)\| = 0 \quad (4.4.2.13)$$

and (4.4.2.12) can therefore be restated as:

$$\lim_{k \rightarrow \infty} \frac{[\tilde{\Theta}^T(k-d) \underline{\phi}(k-d) + d(k)]^2 - [1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)] R^2(k)}{1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)} = 0 \quad (4.4.2.14)$$

which is the desired result, and is valid when $|g(k)| \geq [1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)]^{1/2} R(k)$.

4.5 Adaptive Controller Development Without Unmodeled Dynamics

An adaptive control strategy will now be developed for the system of Section 4.1.1 which is consistent with the controller design of the preceding two chapters. As pointed out in Section 4.1.2, a parameter estimation scheme with proven characteristics will be combined with a proposed controller structure using the Key Technical Lemma to show that the overall system meets certain boundedness and convergence properties.

4.5.1 No Disturbance Case

In this section, the system of (4.1.1.8) with $d(k) = 0$ will be used. The object is to specify a control input at sampling instant k which will result in the system state eventually reaching and remaining on the sliding surface.

First express the system (4.1.1.8) using the definitions of $\underline{\Theta}(k)$ and $\underline{\phi}(k)$ given in (4.1.1.6) and (4.1.1.7):

$$\underline{e}^T \underline{x}(k+d) = \underline{e}^T \underline{A}' \underline{x}(k) + \underline{\Theta}_f \underline{\phi}_f(k) + \sum_{i=1}^d \left\{ \underline{\Theta}_{b_i}^T \underline{\Theta}_{b_i}(k) z^{-(i-1)} \right\} u(k) \quad (4.5.1.1)$$

The current control input can be isolated by rewriting (4.5.1.1):

$$\underline{e}^T \underline{x}(k+d) = \underline{e}^T \underline{A}' \underline{x}(k) + \underline{\Theta}_f \underline{\phi}_f(k) + \underline{\Theta}_{b_1}^T \underline{\phi}_{b_1} u(k) + \sum_{i=2}^d \left\{ \underline{\Theta}_{b_i}^T \underline{\phi}_{b_i}(k) z^{-(i-1)} \right\} u(k) \quad (4.5.1.2)$$

To make $s(k+d) = 0$ requires a $u(k)$ which will cause $\underline{e}^T \underline{x}(k+d) = \underline{e}^T \underline{x}_d(k+d)$. This control input can be calculated from (4.5.1.2):

$$u(k) = \frac{1}{\underline{\Theta}_{b_1}^T \underline{\phi}_{b_1}(k)} \left\{ -\underline{\Theta}_f \underline{\phi}_f(k) - \underline{e}^T \underline{A}' \underline{x}(k) + \right. \\ \left. - \sum_{i=2}^d \left\{ \underline{\Theta}_{b_i}^T \underline{\phi}_{b_i}(k) z^{-(i-1)} \right\} u(k) + \underline{e}^T \underline{x}_d(k+d) \right\} \quad (4.5.1.3)$$

where it is assumed that $\underline{\Theta}_{b_1}^T \underline{\phi}_{b_1}(k) \neq 0$. Note that (4.5.1.3) can be rewritten as:

$$\sum_{i=1}^d \left\{ \underline{\Theta}_{b_i}^T \underline{\phi}_{b_i}(k) z^{-(i-1)} \right\} u(k) = -\underline{\Theta}_f \underline{\phi}_f(k) - \underline{e}^T \underline{A}' \underline{x}(k) + \underline{e}^T \underline{x}_d(k+d) \quad (4.5.1.4)$$

It is clear that Assumption 3 of Section 4.1.1 is necessary in order to keep (4.5.1.4) “stably invertible” from $u(k)$ so that $u(k)$ remains bounded.

Now since parameter vectors $\underline{\Theta}_f$ and $\underline{\Theta}_{b_i}$ are in general unknown, their estimates are used in formulating $u(k)$ (the “certainty equivalence method”). This results in the control law:

$$u(k) = \frac{1}{\hat{\underline{\Theta}}_{b_1}^T(k) \underline{\phi}_{b_1}(k)} \left\{ -\hat{\underline{\Theta}}_f^T(k) \underline{\phi}_f(k) - \underline{e}^T \underline{A}' \underline{x}(k) + \right. \\ \left. - \sum_{i=2}^d \left\{ \hat{\underline{\Theta}}_{b_i}^T(k) \underline{\phi}_{b_i}(k) z^{-(i-1)} \right\} u(k) + \underline{e}^T \underline{x}_d(k+d) \right\} \quad (4.5.1.5)$$

where it is again assumed that $\hat{\underline{\Theta}}_{b_1}^T(k) \underline{\phi}_{b_1}(k) \neq 0$.

Before proceeding with the formal boundedness and convergence proofs, it is helpful to examine the motivation of using a certainty equivalence control scheme coupled with a parameter estimator. Recall that in the parameter estimation problem, $\underline{e}^T \underline{x}(k)$ is predicted using the current estimates for $\underline{\Theta}$:

$$\underline{e}^T \hat{\underline{x}}(k) = \underline{e}^T \underline{A}' \underline{x}(k-d) + \hat{\underline{\Theta}}^T(k-1) \underline{\phi}(k-d) \quad (4.5.1.6)$$

and the parameter estimator is driven by:

$$\underline{e}^T \hat{\hat{\underline{x}}}(k) = \underline{e}^T \underline{x}(k) - \underline{e}^T \hat{\underline{x}}(k) \quad (4.5.1.7)$$

If the prediction error converges to zero, i.e. if:

$$\lim_{k \rightarrow \infty} \underline{e}^T \hat{\hat{\underline{x}}}(k) = 0 \quad (4.5.1.8)$$

then this implies:

$$\lim_{k \rightarrow \infty} \underline{e}^T \hat{\underline{x}}(k) = \underline{e}^T \underline{x}(k) \quad (4.5.1.9)$$

Using the definitions of $\underline{\Theta}$ and $\underline{\phi}$ from (4.1.1.6) and (4.1.1.7), the control law of (4.5.1.5) reduces to:

$$\hat{\underline{\Theta}}^T(k-d) \underline{\phi}(k-d) = \underline{e}^T \underline{x}_d(k) - \underline{e}^T \underline{A}' \underline{x}(k-d) \quad (4.5.1.10)$$

From Section 4.2, each parameter estimation methods results in:

$$\lim_{k \rightarrow \infty} \|\tilde{\underline{\Theta}}(k) - \tilde{\underline{\Theta}}(k-d)\| = 0 \quad (4.5.1.11)$$

Using this result with (4.5.1.10) gives, in the limit as $k \rightarrow \infty$:

$$\hat{\underline{\Theta}}^T(k-1)\underline{\phi}(k-d) = \underline{e}^T \underline{x}_d(k) - \underline{e}'^T \underline{A}' \underline{x}(k-d) \quad (4.5.1.12)$$

Substituting (4.5.1.12) into (4.5.1.6):

$$\lim_{k \rightarrow \infty} \underline{e}^T \hat{\underline{x}}(k) = \underline{e}^T \underline{x}_d(k) \quad (4.5.1.13)$$

But if the prediction error converges to zero, then from (4.5.1.9) and (4.5.1.13) this implies:

$$\lim_{k \rightarrow \infty} \underline{e}^T \underline{x}(k) = \underline{e}^T \underline{x}_d(k) \quad (4.5.1.14)$$

which is the desired result from a control perspective. Thus the objective of the parameter estimation problem (4.5.1.8) simultaneously achieves the objective of the adaptive control problem (4.5.1.14).

4.5.1.1. Normalized Gradient Parameter Estimation

It is now necessary to show that, using the system of (4.5.1.1), the control law of (4.5.1.5), and the normalized gradient parameter estimator of Section 4.2.1, the closed loop system results in a bounded state vector and control input as well as achieving the control objective. As discussed earlier, this will be accomplished using the Key Technical Lemma of Section 4.1.2.

The system of (4.1.1.8) can be written as:

$$\underline{e}^T \underline{x}(k) = \underline{e}'^T \underline{A}' \underline{x}(k-d) + \underline{\Theta}^T \underline{\phi}(k-d) \quad (4.5.1.1.1)$$

From (4.5.1.10), the control law of (4.5.1.5) reduces to:

$$\underline{e}^T \underline{x}_d(k) = \underline{e}^T \underline{A}' \underline{x}(k-d) + \hat{\underline{\Theta}}^T(k-d) \underline{\phi}(k-d) \quad (4.5.1.1.2)$$

Subtracting (4.5.1.12) from (4.5.1.1.1) and using the definition of $s(k)$ gives the prediction error:

$$s(k) = \tilde{\underline{\Theta}}^T(k-d) \underline{\phi}(k-d) \quad (4.5.1.1.3)$$

(4.5.1.1.3) can be substituted into (4.2.1.14) to give:

$$\lim_{k \rightarrow \infty} \frac{s^2(k)}{\underline{\phi}^T(k-d) \underline{\phi}(k-d) + c} = 0 \quad (4.5.1.14)$$

Condition (1) of the lemma is therefore satisfied, with $p(k) = s(k)$, $b_1(k) = c$, $b_2(k) = 1$, and $\underline{\sigma}(k) = \underline{\phi}(k-d)$. This choice of $b_1(k)$ and $b_2(k)$ also satisfies Condition (2).

To show that Condition (3) of the lemma is satisfied, it is necessary that $s(k)$ linearly bounds $\underline{\phi}(k-d)$. This can be done by first showing that $\underline{x}(k)$ linearly bounds $\underline{\phi}(k-d)$. Notice that $\underline{\phi}(k-d)$ is made up of functions of the state $\underline{\phi}_f$ and $\underline{\phi}_{b_i}$, $1 \leq i \leq d$ and of the control input $u(i)$, $k-2d < i \leq k-d$. Since the parameter estimates remain bounded due to the boundedness properties of the parameter estimator, each $u(i)$ is bounded if $\underline{x}(k-d)$ and $\underline{\phi}(k-d)$ are bounded from Assumption (3) of Section 4.1.1. Assumption (4) of Section 4.1.1 requires that $\underline{x}(k-d)$ linearly bound $\underline{\phi}_f$ and that $\underline{\phi}_{b_i}$ be linearly bounded by a constant, and thus $\underline{x}(k)$ linearly bounds each component of $\underline{\phi}(k-d)$.

To demonstrate that $s(k)$ linearly bounds $\underline{x}(k)$, recall that the selection of vector \underline{e} in Chapter 2 established the ability to set the dynamics on the sliding surface as desired. From Section 2.4.2, this implies the system error dynamics:

$$\tilde{\underline{x}}(k+1) = \begin{bmatrix} \underline{A}' \\ -\underline{e}'^T \underline{A}' \end{bmatrix} \tilde{\underline{x}}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} s(k) \quad (4.5.1.15)$$

where $s(k)$ represents the distance from the surface. Assuming that the dynamics were selected to be asymptotically stable in (4.5.1.1.5), then if $s(k)$ is bounded, $\tilde{\underline{x}}(k)$ will be bounded as well. Since $\underline{x}_d(k)$ is assumed to be bounded, then this implies that $\underline{x}(k)$ will also be bounded. Conversely, if $s(k)$ is unbounded, then $\tilde{\underline{x}}(k)$ will be also. Thus, because the dynamics are linear in (4.5.1.1.5), $s(k)$ can always linearly bound $\tilde{\underline{x}}(k)$ (and hence $\underline{x}(k)$).

With all conditions of the Key Technical Lemma satisfied, this allows for the results that $\underline{\phi}(k-d)$ is bounded for all k and that $\lim_{k \rightarrow \infty} s(k) = 0$. The first result shows that the control input remains bounded for all k , and the second result implies that the surface is eventually reached which results in the asymptotic convergence of the tracking error to zero and the boundedness of the state vector.

This is a strong result when compared to the non-adaptive controllers of Chapter 3 in that the tracking error can be shown to converge to zero without any restrictions on the initial parameter errors. As noted earlier, convergence of the tracking error to zero does not imply that the parameter error vector converges to zero. This point will be discussed further in Section 4.5.1.3.

4.5.1.2 Least Squares Parameter Estimation

As in Section 4.5.1.1, it is necessary to show that, using the system of (4.5.1.1), the control law of (4.5.1.5) and the least squares parameter estimators of Section 4.2.2, the closed loop system results in a bounded state vector and control input as well as achieving the control objective.

Exactly as in Section 4.5.1.1, Condition (1) of the Key Technical Lemma is established by substitution of (4.5.1.1.3) into (4.2.2.1.21):

$$\lim_{k \rightarrow \infty} \frac{s^2(k)}{1 + \underline{\phi}^T(k-d) \underline{P}(k-d-1) \underline{\phi}(k-d)} = 0 \quad (4.5.1.2.1)$$

Since $\underline{P}(k-d-1)$ is positive definite for all k and upper bounded by the $\underline{P}(-1)$ of (4.2.2.1.3), (4.5.1.2.1) implies:

$$\lim_{k \rightarrow \infty} \frac{s^2(k)}{1 + \alpha \underline{\phi}^T(k-d) \underline{\phi}(k-d)} = 0 \quad (4.5.1.2.2)$$

where $0 < \alpha < \epsilon$ (from 4.2.2.1.3). Condition (1) is therefore satisfied with $p(k) = s(k)$, $b_1(k) = 1$, $b_2(k) = \alpha$, and $\underline{\sigma}(k) = \underline{\phi}(k-d)$. Condition (2) is also satisfied with this choice of $b_1(k)$ and $b_2(k)$. Condition (3) is also satisfied exactly as in Section 4.5.1.1. This results in a bounded $\underline{\phi}(k-d)$ and $s(k)$ converging to zero as before.

When using the least squares algorithms of Section 4.2.2, it is important that $\underline{P}(k-d-1)$ remain bounded away from zero (i.e. remain positive definite rather than positive semi-definite) so that α remains > 0 . This can be accomplished by stopping the updating on $\underline{P}(k-d-1)$ when it reaches some lower bound, which effectively changes the algorithm into a normalized gradient estimator.

4.5.1.3 Parameter Convergence

Parameter convergence was discussed in Section 4.2.3 for the parameter estimation problem, and was shown to depend on the characteristics of the vector $\underline{\phi}(k)$. In the adaptive control case, the vector $\underline{\phi}(k)$ is composed of functions of the state and of the control input. These signals are not under the direct control of the experimenter, however, since only the input $u(k)$ is deterministic. Furthermore, $u(k)$ may not be available for manipulation directly because it may for example, be required to be specified in

order for closed loop system stability to be maintained. It would thus appear that only $\underline{x}_d(k)$ can be easily varied to achieve a specified behavior in $\underline{\phi}(k)$.

For linear systems, it can be shown how conditions on the input $u(k)$ can lead to parameter convergence under certain assumptions on the system type. In this case, $\underline{x}_d(k)$ could be specified in this case to achieve the requisite condition for $u(k)$ (normally a "sufficient richness" or "persistent excitation" condition specified in the frequency domain). For nonlinear systems, however, proving an analagous result may not be possible. For these systems, a reasonable choice of $\underline{x}_d(k)$ is to require it to meet the same conditions as that for a linear system.

In Section 4.5.1 it was shown how, with the proper selection of control law, $\lim_{k \rightarrow \infty} s(k) = 0$. By the proper selection of \underline{e} in Chapter 2, it was shown that this implies $\underline{x}(k) \rightarrow \underline{x}_d(k)$. This further implies that $\underline{\phi}(\underline{x}(k), k) \rightarrow \underline{\phi}(\underline{x}_d(k), k)$ since $\underline{\phi}$ is composed of functions of the state and of the input (which is also a function of the state). The difficulty occurs when trying to specify $\underline{x}_d(k)$ such to make $\underline{\phi}(\underline{x}_d(k), k)$ "sufficiently rich" or "persistently exciting", since it is, in general, a nonlinear function of $\underline{x}_d(k)$. Some current research in this area may be found in [73-77].

Example 4.1

Let the true system be given by:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.22 \\ 0 & 0.95 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{0.0066}{1+0.05|x_2(k)|} \end{bmatrix} u(k) \quad (\text{Ex. 4.1.1})$$

and the available model by:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.22 \\ 0 & f' \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{b'}{1+0.05|x_2(k)|} \end{bmatrix} u(k) \quad (\text{Ex. 4.1.2})$$

with $f' = 0.8$ and $b' = 0.0082$ as in Example 3.3, with parameter constraints $0.65 \leq f' \leq 0.95$ and $0.0066 \leq b' \leq 0.0103$. The model (Ex. 4.1.2) can be put in the form of (4.1.1.3), with:

$$\underline{A}' = [1 \quad 0.22] \quad (\text{Ex. 4.1.3})$$

$$\hat{\underline{\Theta}}_f(k=0) = [0.8] \quad (\text{Ex. 4.1.4})$$

$$\underline{\phi}_f(k) = [x_2(k)] \quad (\text{Ex. 4.1.5})$$

$$\hat{\underline{\Theta}}_b(k=0) = [0.0082] \quad (\text{Ex. 4.1.6})$$

$$\underline{\phi}_b(k) = \left[\frac{1}{1 + 0.05 |x_2(k)|} \right] \quad (\text{Ex. 4.1.7})$$

and all the conditions of Section 4.1.1 are seen to be met.

Using the definitions (4.1.1.6) and (4.1.1.7):

$$\underline{\Theta} = [0.95 \quad 0.0066]^T \quad (\text{Ex. 4.1.8})$$

$$\underline{\phi}(k) = \left[x_2(k) \quad \frac{u(k)}{1 + 0.05 |x_2(k)|} \right] \quad (\text{Ex. 4.1.9})$$

and:

$$\hat{\underline{\Theta}}(k=0) = [0.08 \quad 0.0082]^T \quad (\text{Ex. 4.1.10})$$

The corresponding scalar system is:

$$\underline{e}^T(k+1) = \underline{e}^T \underline{A}' \underline{x}(k) + \underline{\Theta}^T \underline{\phi}(k) \quad (\text{Ex. 4.1.11})$$

where:

$$\underline{e} = [0.5 \quad 1.0]^T \quad (\text{Ex. 4.1.12})$$

$$\underline{e}' = [0.5] \quad (\text{Ex. 4.1.13})$$

which assigns an eigenvalue of $p = 0.89$ to the dynamics on the surface.

Before applying the parameter estimator to (Ex. 4.1.11), notice that the parameter values of (Ex. 4.1.10) are approximately two orders of magnitude apart. To enhance the numerical conditioning of the parameter estimator and to ensure that the parameters are given approximately the same emphasis during adaptation, the parameter vector can be rescaled by letting:

$$\underline{\phi}(k) = [x_2(k) \quad \frac{u(k)}{100\{1 + 0.05 |x_2(k)|\}}] \quad (\text{Ex. 4.1.14})$$

$$\hat{\underline{\Theta}}(k=0) = [0.08 \quad 0.82]^T \quad (\text{Ex. 4.1.15})$$

This technique can also be used to emphasize or de-emphasize the adaptation of a particular parameter. The parameter constraints become $0.65 \leq \hat{\Theta}_1(k) \leq 0.95$ and $0.66 \leq \hat{\Theta}_2(k) \leq 1.03$.

Using the normalized gradient parameter estimator of (4.2.1.1):

$$\hat{\underline{\Theta}}(k) = \hat{\underline{\Theta}}(k-1) + \frac{a(k)\underline{\phi}(k-1)[\underline{e}^T \underline{x}(k) - \underline{e}^T \underline{A}' \underline{x}(k) - \hat{\underline{\Theta}}^T(k-1)\underline{\phi}(k-1)]}{\underline{\phi}^T(k-1)\underline{\phi}(k-1) + c} \quad (\text{Ex. 4.1.16})$$

where $a(k) = 1$ and $c = 0.01$. As in Section 4.3.1, the parameter estimate is changed to the upper or lower constraint, as appropriate, if the estimator specifies a value outside of the constrained limits.

The control law is specified in (4.5.1.5):

$$u(k) = \frac{100\{1 + 0.05 |x_2(k)|\}}{\hat{\Theta}_2(k)} \{-\hat{\Theta}_1(k)x_2(k) - \underline{e}^T \underline{A}' \underline{x}(k) + \underline{e}^T \underline{x}_d(k+1)\} \quad (\text{Ex. 4.1.17})$$

A simulation was performed using the desired trajectory for $x_{2_d}(k)$:

$$x_{2_d}(k) = 10\sin(0.033k) - 10\sin(0.066k)$$

(Ex. 4.1.18)

The trajectory consistency requirement is identical to that of (Ex. 2.2.4). The system behavior is summarized in the s trajectory of Figure 4.1. As predicted, s converges to the surface as time progresses and is seen to be essentially zero after approximately 100 sampling periods.

The parameter estimator performance is summarized by the parameter error norm (the Liapunov function value of (4.1.1.2)) of Figure 4.2. The norm is non-increasing and is essentially reduced to zero after about 35 sampling periods. The individual parameter estimates are shown in Figures 4.3 and 4.4, where “parameter 1” corresponds to f' and “parameter 2” to $100b'$. The known parameter constraints have been used in accordance with the guidance provided in Section 4.3.1.

Concerning the matter of parameter convergence, the desired trajectory was selected as two different frequency sinusoids in order to achieve a “sufficiently rich” regressor vector as specified in Sections 4.2.3 and 4.5.1.3. Because of the nonlinear nature of the system (Ex. 4.1.1), however, this does not guarantee the sufficient richness property. The minimum eigenvalue of the summed regressor matrix $\Phi(k)\Phi^T(k)$ (see (4.2.3.1)) was therefore calculated and is shown in Figure 4.5. This value is seen to increase steadily throughout the simulation run which implies that the parameter convergence should be asymptotic. This behavior is confirmed in Figure 4.2.

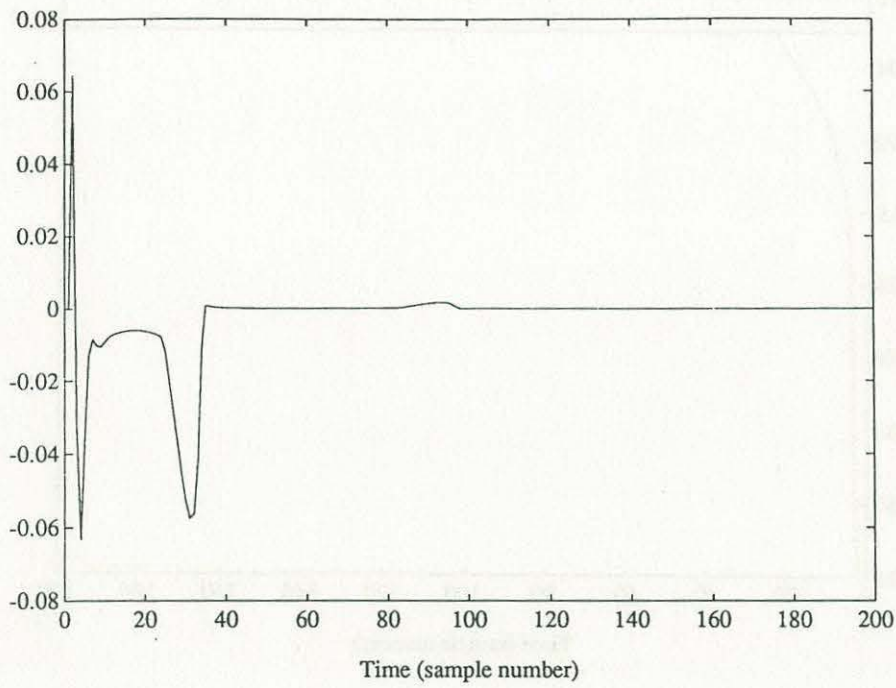


Figure 4.1 S trajectory

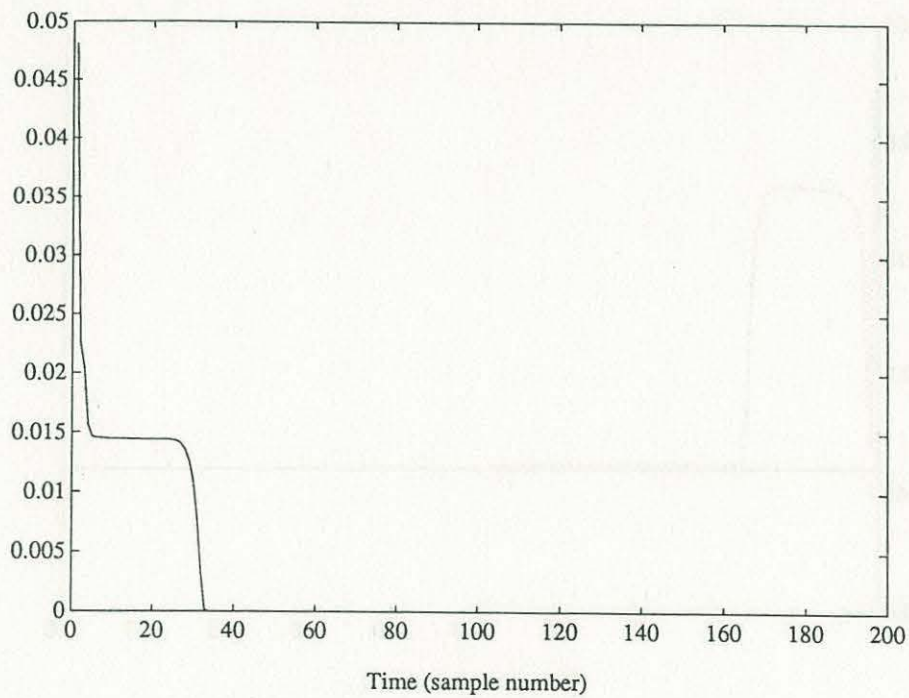


Figure 4.2 Parameter error norm

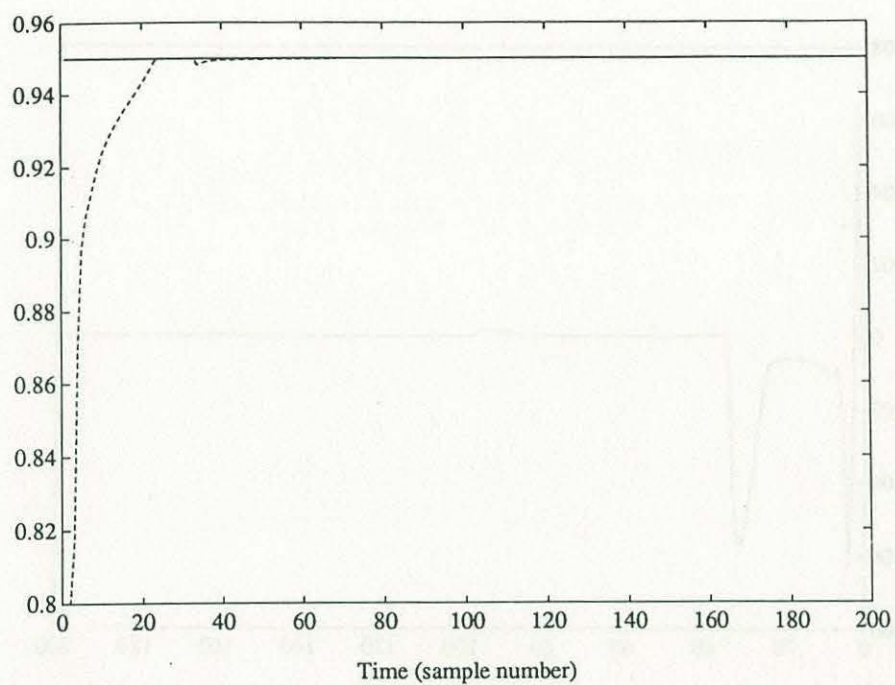


Figure 4.3 Actual (-) and estimated (- -) parameter 1

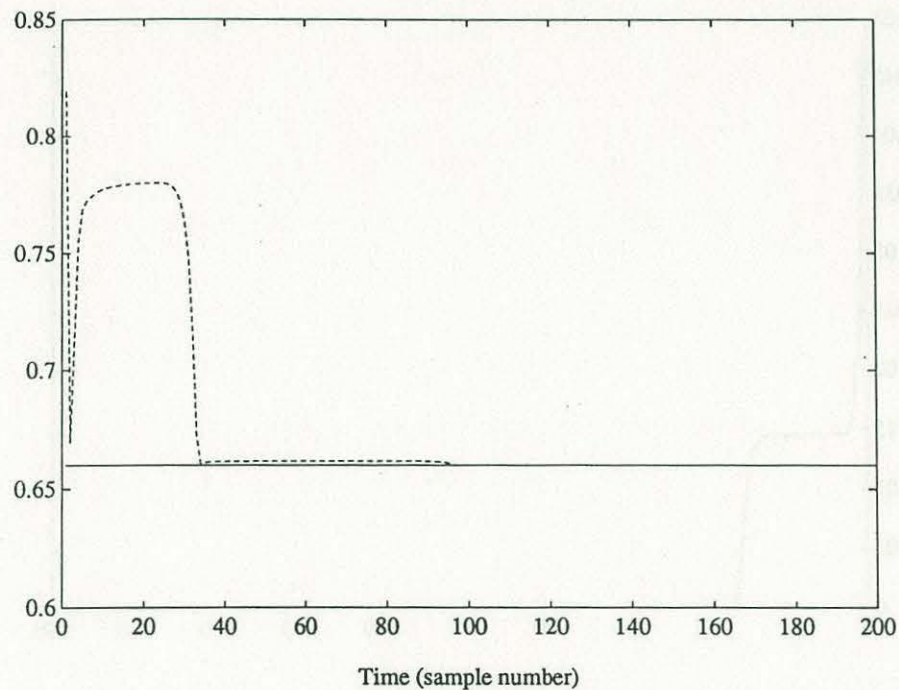


Figure 4.4 Actual (-) and estimated (- -) parameter 2

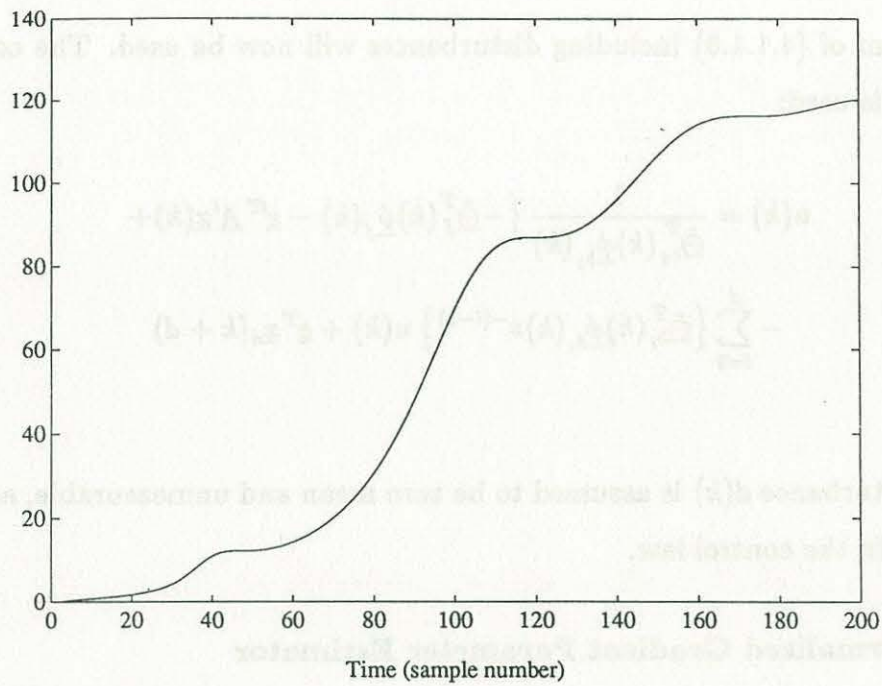


Figure 4.5 Minimum eigenvalue of the regressor matrix

4.5.2 Extension to Bounded Disturbances

The system of (4.1.1.8) including disturbances will now be used. The control law of (4.5.1.1.5) is used:

$$u(k) = \frac{1}{\hat{\Theta}_{b_1}^T(k)\underline{\phi}_{b_1}(k)} \left\{ -\hat{\Theta}_f^T(k)\underline{\phi}_f(k) - \underline{e}^T \underline{A}' \underline{x}(k) + \right. \\ \left. - \sum_{i=2}^d \left\{ \hat{\Theta}_{b_i}^T(k)\underline{\phi}_{b_i}(k)z^{-(i-1)} \right\} u(k) + \underline{e}^T \underline{x}_d(k+d) \right\} \quad (4.5.2.1)$$

where the disturbance $d(k)$ is assumed to be zero mean and unmeasurable, and thus is not included in the control law.

4.5.2.1 Normalized Gradient Parameter Estimator

As in Section 4.5.1.1, it is necessary to show that, using the system of (4.1.1.8), the control law of (4.5.2.1) and the normalized gradient parameter estimator with bounded disturbances of Section 4.4.1, the closed loop system results in a bounded state vector and control input as well as achieving the control objective.

The system of (4.1.1.8) can be written as:

$$\underline{e}^T \underline{x}(k) = \underline{e}^T \underline{A}' \underline{x}(k-d) + \underline{\Theta}^T \underline{\phi}(k-d) + d(k-d) \quad (4.5.2.1.1)$$

Using the definitions of $\underline{\Theta}$ and $\underline{\phi}$ from (4.1.1.6) and (4.1.1.7), the control law of (4.5.2.1) reduces to:

$$\underline{e}^T \underline{x}_d(k) = \underline{e}^T \underline{A}' \underline{x}(k-d) + \hat{\underline{\Theta}}^T(k-d)\underline{\phi}(k-d) \quad (4.5.2.1.2)$$

Subtracting (4.5.2.1.2) from (4.5.2.1.1) and using the definition of $s(k)$ gives the prediction error:

$$s(k) = \tilde{\underline{\Theta}}^T(k-d)\underline{\phi}(k-d) + d(k-d)$$

(4.5.2.1.3)

(4.5.2.1.3) can be substituted into (4.4.1.1.8) to give:

$$\lim_{k \rightarrow \infty} \frac{[s(k) - R(k)\text{sgn}\{g(k)\}]^2}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} = 0$$

(4.5.2.1.4)

From (4.4.1.6) and (4.5.2.1.2):

$$\lim_{k \rightarrow \infty} g(k) = s(k)$$

(4.5.2.1.5)

and so (4.5.2.1.4) becomes:

$$\lim_{k \rightarrow \infty} \frac{[s(k) - R(k)\text{sgn}\{s(k)\}]^2}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} = 0$$

(4.5.2.1.6)

which is valid when $|s(k)| \geq R(k)$. Condition (1) of the lemma is therefore satisfied, with $p(k) = s(k) - R(k)\text{sgn}\{s(k)\}$, $b_1(k) = c$, $b_2(k) = 1$, and $\underline{\sigma}(k) = \underline{\phi}(k-d)$. This choice of $b_1(k)$ and $b_2(k)$ also satisfies Condition (2). Condition (3) is satisfied exactly as in Section 4.5.1.1 since $R(k)$ is bounded.

This allows for the result that $\underline{\phi}(k-d)$ is bounded for all k and $\lim_{k \rightarrow \infty} s(k) - R(k)\text{sgn}\{s(k)\} = 0$. The first result shows that the control input remains bounded, and the second result implies that s converges to within a bounded distance $R(k)$ from the surface.

The control law of (4.5.2.1) can easily be modified to make it compatible with the control laws of Chapter 3 which utilize the boundary layer. Recall the control law given in (3.1.5) for use with bounded disturbances:

$$u(k) = \frac{1}{b} \left\{ s(k) - f - \underline{e}^T \underline{A}' \underline{x}(k) + \underline{e}^T \underline{x}_d(k+1) - K \text{sat} \left\{ \frac{s(k)}{\Phi} \right\} \right\} \quad (4.5.2.1.7)$$

In the notation of this section, (4.5.2.1.7)

$$u(k) = \frac{1}{\hat{\underline{\Theta}}_{b_1}^T(k) \underline{\phi}_{b_1}(k)} \left\{ s(k) - \hat{\underline{\Theta}}_f^T(k) \underline{\phi}_f(k) - \underline{e}^T \underline{A}' \underline{x}(k) + \right. \\ \left. - \sum_{i=1}^d \left\{ \hat{\underline{\Theta}}_{b_i}^T(k) \underline{\phi}_{b_i}(k) z^{-(i-1)} \right\} u(k) + \underline{e}^T \underline{x}_d(k+d) - K \text{sat} \left\{ \frac{s(k)}{\Phi} \right\} \right\} \quad (4.5.2.1.8)$$

The boundary layer Φ was selected such that the eigenvalue governing the s dynamics was set to p :

$$1 - \frac{K}{\Phi} = p \quad (4.5.2.1.9)$$

and from (3.1.4), K was selected as:

$$K = D + \eta \quad (4.5.2.1.10)$$

From (4.5.2.1.9) and (4.5.2.1.10):

$$\Phi = \frac{D + \eta}{1 - p} \quad (4.5.2.1.11)$$

A reasonable criteria to use in shifting from the non-adaptive to adaptive controller is whether s is inside or outside the boundary layer. That is, when inside the boundary layer, the control law of (4.5.2.1.8) is used and adaptation does not occur. Outside the boundary layer, the adaptive control law is used and adaptation takes place. The control laws should therefore be continuous at the boundary layer edge. For this purpose, note that the non-adaptive control law of (4.5.2.1.8) takes on the following value at the boundary layer edge (i.e., at $s(k) = \Phi \text{sgn}\{s(k)\}$):

$$u(k) = \frac{1}{\hat{\underline{\Theta}}_{b_1}^T(k)\underline{\phi}_{b_1}(k)} \left\{ -\hat{\underline{\Theta}}_f^T(k)\underline{\phi}_f(k) - \underline{e}^T \underline{A}' \underline{x}(k) + \right. \\ \left. - \sum_{i=1}^d \left\{ \hat{\underline{\Theta}}_{b_i}^T(k)\underline{\phi}_{b_i}(k)z^{-(i-1)} \right\} u(k) + \underline{e}^T \underline{x}_d(k+d) + p\Phi \text{sgn}\{s(k)\} \right\} \quad (4.5.2.1.12)$$

The adaptive control law of (4.5.2.1) can be made equivalent to the non-adaptive control law (4.5.2.1.12) at the boundary layer edge by augmenting it with the term $p\Phi \text{sgn}\{s(k)\}$. The augmented control law reduces to:

$$\underline{e}^T \underline{x}_d(k+d) = -p\Phi \text{sgn}\{s(k-d)\} + \underline{e}^T \underline{A}' \underline{x}(k-d) + \hat{\underline{\Theta}}^T(k-d)\underline{\phi}(k-d) \quad (4.5.2.1.13)$$

Subtracting this from (4.5.2.1.1) gives the new prediction error:

$$s(k) = p\Phi \text{sgn}\{s(k-d)\} + \tilde{\underline{\Theta}}^T(k-d)\underline{\phi}(k-d) + d(k-d) \quad (4.5.2.1.14)$$

(4.5.2.1.14) can be substituted into (4.4.1.18) to get:

$$\lim_{k \rightarrow \infty} \frac{[s(k) - p\Phi \text{sgn}\{s(k-d)\} - R(k)\text{sgn}\{g(k)\}]^2}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} = 0 \quad (4.5.2.1.15)$$

From (4.4.1.6) and (4.5.2.1.13):

$$g(k) = s(k) - p\Phi \text{sgn}\{s(k-d)\} \quad (4.5.2.1.16)$$

Now select $R(k)$ as :

$$R(k) = (1 - p)\Phi \quad (4.5.2.1.17)$$

From (4.5.2.1.11), this guarantees that $R(k) \geq D$ as required. The condition on adaptation remains as $|g(k)| \geq R(k)$, which from (4.5.2.1.16) and (4.5.2.1.17) is met if $|s(k)| \geq \Phi$ (being outside the boundary layer). With this condition satisfied, from (4.5.2.1.16) it can be seen that $\text{sgn}\{g(k)\} = \text{sgn}\{s(k)\}$, and so using (4.5.2.1.17), (4.5.2.1.15) becomes:

$$\lim_{k \rightarrow \infty} \frac{[s(k) - \underline{\phi} \text{sgn}\{s(k-d)\}]^2}{\underline{\phi}^T(k-d)\underline{\phi}(k-d) + c} = 0 \quad (4.5.2.1.18)$$

Note that the augmented control law of (4.5.2.1.12) again satisfies the conditions of the Key Technical Lemma, and thus keeps the control input bounded and leads s to within a bounded region of the surface, which in turn keeps the state vector bounded.

The use of a time-variant boundary layer can be accommodated as well, and can be used to account for time-variant disturbances and modeling errors.

4.5.2.2 Least Squares Parameter Estimator

The development for the least squares estimator follows that of the normalized gradient estimator in the previous section. Condition (1) is satisfied by substituting (4.5.2.1.3) into (4.4.2.1.4) to get:

$$\lim_{k \rightarrow \infty} \frac{s^2(k) - [1 + \underline{\phi}^T(k-d)P(k-d-1)\underline{\phi}(k-d)]R^2(k)}{1 + \alpha \underline{\phi}^T(k-d)\underline{\phi}(k-d)} = 0 \quad (4.5.2.2.1)$$

where $0 < \alpha < \epsilon$ (from 4.2.2.1.3). Condition (2) is also satisfied with this choice of $b_1(k)$ and $b_2(k)$. The boundedness of $R(k)$ allows for Condition (3) of the lemma to

be met exactly as in Section 4.5.2.1. As stressed in Section 4.5.1.2, it is necessary for $\underline{P}(k-d-1)$ to remain bounded away from zero for the analysis to hold.

The result is that $s(k)$ will converge to the region $\pm[1 + \phi^T(k-d)\underline{P}(k-d-1)\phi(k-d)]R(k)$. $\underline{P}(k-d-1)$ will decrease for any $\phi(k-d) \neq 0$, and thus $s(k)$ will converge to a region whose size is governed by $\underline{P}(k=\infty)$ as well as by $\phi(k-d)$. Since the lemma states that $\phi(k-d)$ is bounded, $\underline{P}(k-d-1)$ can decrease to an arbitrarily small value which implies that $s(k)$ converges to $\pm R(k)$. The modified control law of (4.5.1.1.12) can be used as before to accomodate the use of the boundary layer with the non-adaptive controllers of Chapter 3.

Example 4.2

The system of (Ex. 4.1.1) is now considered with an external disturbance:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.22 \\ 0 & 0.95 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{0.0066}{1+0.05|x_2(k)|} \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d(k) \quad (\text{Ex. 4.2.1})$$

where $|d(k)| \leq 0.2$. The available model is again given by (Ex. 4.1.2):

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.22 \\ 0 & f' \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{b'}{1+0.05|x_2(k)|} \end{bmatrix} u(k) \quad (\text{Ex. 4.2.2})$$

with $0.65 \leq f' \leq 0.95$ and $0.0066 \leq b' \leq 0.0103$. The parameter and regressor vectors are scaled as in Example 4.1:

$$\underline{\phi}(k) = [x_2(k) \quad \frac{u(k)}{100\{1 + 0.05 |x_2(k)|\}}] \quad (\text{Ex. 4.2.3})$$

$$\hat{\underline{\theta}}(k=0) = [0.08 \quad 0.82]^T \quad (\text{Ex. 4.2.4})$$

with parameter constraints $0.65 \leq \hat{\theta}_1 \leq 0.95$ and $0.66 \leq \hat{\theta}_2(k) \leq 1.03$.

The control law to be used inside the boundary layer is obtained from (4.5.2.1.8):

$$u(k) = \frac{100\{1 + 0.05 |x_2(k)|\}}{\hat{\theta}_2(k)} \left\{ s(k) - \hat{\theta}_1(k)x_2(k) - \underline{e}^T \underline{A}' \underline{x}(k) + \right. \\ \left. + \underline{e}^T \underline{x}_d(k+1) - K \text{sat} \left\{ \frac{s(k)}{\Phi} \right\} \right\} \quad (\text{Ex. 4.2.5})$$

where:

$$K = D + \eta$$

(Ex. 4.2.6)

and:

$$\Phi = \frac{K}{1 - p}$$

(Ex. 4.2.7)

with $D = 0.2, \eta = 0.1$, and $p = 0.89$ as used in previous examples. Outside the boundary layer, the control law of (4.5.2.1.12) is used:

$$u(k) = \frac{100\{1 + 0.05 |x_2(k)|\}}{\hat{\Theta}_2(k)} \left\{ -\hat{\Theta}_1(k)x_2(k) - \underline{e}'^T \underline{A}' \underline{x}(k) + \right. \\ \left. + \underline{e}^T \underline{x}_d(k+1) + p\Phi \operatorname{sgn}\{s(k)\} \right\}$$

(Ex. 4.2.8)

along with a suitable parameter estimator.

Using the normalized gradient parameter estimator with bounded disturbances (4.4.1.4):

$$\hat{\Theta}(k) = \hat{\Theta}(k-1) + \frac{a(k)\underline{\phi}(k-1)g_{\Delta}(k)}{\underline{\phi}^T(k-1)\underline{\phi}(k-1) + c}$$

(Ex. 4.2.9)

where $a(k) = 1, c = 0.01$, and :

$$g_{\Delta}(k) = \begin{cases} s(k) - \Phi \operatorname{sgn}\{s(k)\} & \text{if } |s(k)| \geq \Phi \\ 0 & \text{otherwise} \end{cases}$$

(Ex. 4.2.10)

A simulation was performed using the desired trajectory for $x_{2_d}(k)$:

$$x_{2_d}(k) = 10\sin(0.033k) - 10\sin(0.066k)$$

(Ex. 4.2.11)

with trajectory consistency requirement (Ex. 2.2.4) and disturbance:

$$d(k) = 0.2\sin(\pi k/2)$$

(Ex. 4.2.12)

The s trajectory is shown in Figure 4.6 along with the boundary layer Φ . The parameter error norm is shown in Figure 4.7. Comparing Figure 4.6 with 4.7 confirms that adaptation only occurs when s moves outside the boundary layer. Furthermore, the non-increasing nature of the parameter error norm verifies that the disturbance term has been properly accounted for in the parameter estimation algorithm.

An important observation to be made is that the use of the boundary layer does not allow for the parameter estimates to converge exactly to their true values as in Example 4.1. This is a tradeoff that must be accepted if the convergent properties of the parameter estimator and stability of the closed loop system are to be guaranteed.

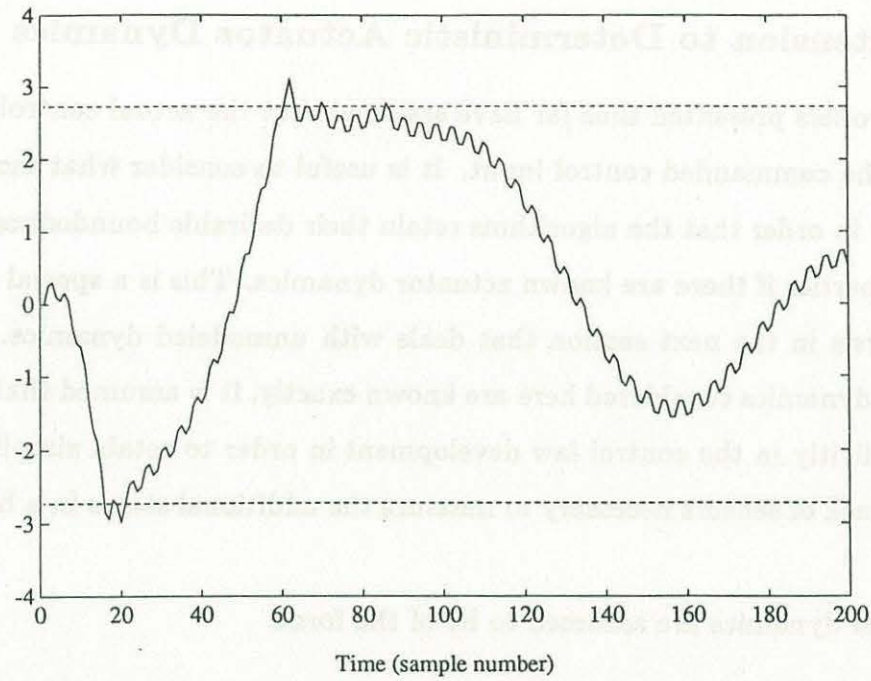


Figure 4.6 $S(-)$ and $\Phi(--)$

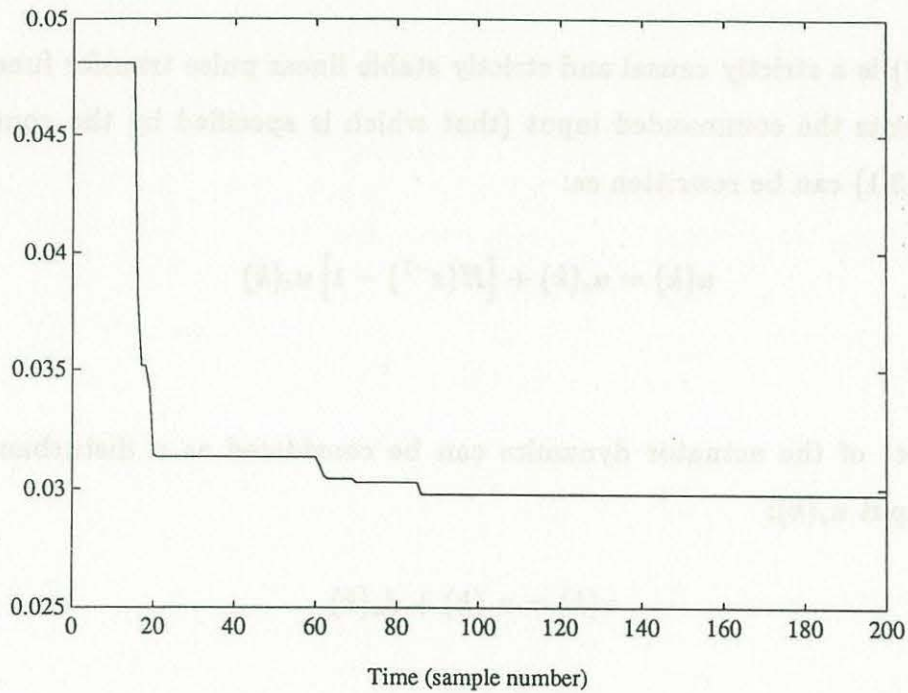


Figure 4.7 Parameter error norm

4.5.3 Extension to Deterministic Actuator Dynamics

The controllers presented thus far have assumed that the actual control input was identical to the commanded control input. It is useful to consider what modifications are necessary in order that the algorithms retain their desirable boundedness and convergence properties if there are known actuator dynamics. This is a special case of the general analysis in the next section that deals with unmodeled dynamics. Although the actuator dynamics considered here are known exactly, it is assumed that they were not used explicitly in the control law development in order to retain simplicity or because of the lack of sensors necessary to measure the additional states in a higher order controller.

The actual dynamics are assumed to be of the form:

$$u(k) = H(z^{-1})u_c(k) \quad (4.5.3.1)$$

where $H(z^{-1})$ is a strictly causal and strictly stable linear pulse transfer function and $u_c(k)$ represents the commanded input (that which is specified by the control algorithm). (4.5.3.1) can be rewritten as:

$$u(k) = u_c(k) + [H(z^{-1}) - 1] u_c(k) \quad (4.5.3.2)$$

and the effect of the actuator dynamics can be considered as a disturbance to the command input $u_c(k)$:

$$u(k) = u_c(k) + d_u(k) \quad (4.5.3.3)$$

where:

$$d_u(k) = [H(z^{-1}) - 1] u_c(k) \quad (4.5.3.4)$$

This can be incorporated into the system description (4.1.1.8):

$$\underline{e}^T \underline{x}(k+d) = \underline{\Theta}^T \underline{\phi}(k) + d(k) \quad (4.5.3.5)$$

as:

$$d(k) = \sum_{i=1}^d \left\{ \underline{\Theta}_{b_i}^T \underline{\phi}_{b_i}(k) z^{-(i-1)} \right\} d_u(k) \quad (4.5.3.6)$$

The methods of Section 4.5.2 can now be used to formulate the control algorithm, where it is necessary to pick $D(k)$ such that:

$$D(k) \geq \left| \sum_{i=1}^d \left\{ \underline{\Theta}_{b_i}^T \underline{\phi}_{b_i}(k-d) z^{-(i-d-1)} \right\} d_u(k) \right| \quad (4.5.3.7)$$

Since $\underline{\Theta}_{b_i}$ is in general unknown, it is necessary to select a $D(k)$ which will sufficiently bound the above expression when $\hat{\underline{\Theta}}_{b_i}^T(k)$ is used. Note that the boundedness of $D(k)$ will be assured if $u_c(k)$ remains bounded because of the stability of $H(z^{-1})$.

The controller structures of Section 4.5.2 can now be applied to the system of (4.1.1.8) where the disturbance term is augmented with that of (4.5.3.6). This will maintain the convergence properties of the parameter estimator. As will be discussed in Chapter 5, the selection of control bandwidth must be tempered by the non-deterministic disturbance environment and the unmodeled dynamics present in the system in order to ensure closed loop system stability.

4.6 Adaptive Control in the Presence of Unmodeled Dynamics

The adaptive controllers proposed in the previous section were designed for the system and assumptions of Section 4.1.1. A key assumption in this section was that the order of the system model was equal to that of the actual system. Unfortunately, this assumption is rarely applicable in the application to physical systems which can always be considered as infinite dimensional on some scale. It is thus necessary to consider the effects of these unmodeled dynamics when a controller designed from a lower order model is used. Because of the wide variety of situations that may be encountered in practice, it is normally more useful to generate a set of general guidelines that are widely applicable rather than base an analysis on highly specific cases. This section will present a generalized analysis and offer suggestions on the use of adaptive controllers for high order systems based on reduced order models.

To maintain the simplicity of notation, the system of (4.1.1.8) will be used with input delay $d = 1$. As in previous sections, the analysis is easily generalized to systems with delay $d > 1$. The system description is thus:

$$\underline{e}^T \underline{x}(k+1) = \underline{e}^T \underline{A}' \underline{x}(k) + \underline{\Theta}^T \underline{\phi}(k) + d(k) \quad (4.6.1)$$

It is to be remembered, however, that (4.6.1) is a reduced order model with the actual system having dimension $> n$.

4.6.1 Potential Mechanisms for Instability

In order to design algorithms based on low order models that will work effectively in a higher order system, it is first necessary to analyze what problems may arise that can lead to poor performance or instability. It is interesting to note that the determination

of instability in the adaptive control case is actually quite difficult. This is because the notion of stability or instability is associated with an equilibrium point (for which there may be many in a nonlinear system). Instability about an equilibrium point can be deduced from a linearized analysis about the point, but this does not imply the global instability of the system, since other equilibrium points or regions may be stable. The highly complicated nonlinear dynamics of the adaptive control system thus make a linearized stability analysis suspect and open to misinterpretation [50].

Two such problems will be analyzed in this section. The first issue is that of bandwidth modification that can occur when the parameter estimator is in operation, and the second is the corruption of signals used in the estimator that degrade its performance.

4.6.1.1 Bandwidth Modification

The analysis of the n dimensional state $\underline{x}(k)$ can be simplified by instead considering the behavior of the summarizing scalar $s(k)$. For this purpose, using the definition of $\tilde{\underline{\Theta}}$, the system of (4.6.1) can be expressed as:

$$\underline{e}^T \underline{x}(k+1) = \underline{e}^T \underline{A}' \underline{x}(k) + \hat{\underline{\Theta}}^T(k) \underline{\phi}(k) + \tilde{\underline{\Theta}}^T(k) \underline{\phi}(k) + d(k) \quad (4.6.1.1.1)$$

From (4.5.2.1.2), the adaptive control law reduces to:

$$\underline{e}^T \underline{x}_d(k+1) = \underline{e}^T \underline{A}' \underline{x}(k) + \hat{\underline{\Theta}}^T(k) \underline{\phi}(k) \quad (4.6.1.1.2)$$

Subtracting (4.6.1.1.2) from (4.6.1.1.1) gives:

$$s(k+1) = \tilde{\underline{\Theta}}^T \underline{\phi}(k) + d(k) \quad (4.6.1.1.3)$$

Now make the definition:

$$\underline{\phi}(k) = \underline{\phi}(k-1) + \Delta\underline{\phi}(k-1) \quad (4.6.1.1.4)$$

Using (4.6.1.1.4), (4.6.1.1.3) becomes:

$$s(k+1) = \tilde{\underline{\Theta}}^T(k)\underline{\phi}(k-1) + \tilde{\underline{\Theta}}^T(k)\Delta\underline{\phi}(k-1) + d(k) \quad (4.6.1.1.5)$$

If the controller uses a normalized gradient parameter estimator, $\tilde{\underline{\Theta}}(k)$ can be obtained from (4.2.1.3) and substituted in (4.6.1.15):

$$s(k+1) = \tilde{\underline{\Theta}}^T(k-1)\underline{\phi}(k-1) - \frac{a(k)\underline{\phi}^T(k-1)\underline{\phi}(k-1)\tilde{\underline{\Theta}}^T(k-1)\underline{\phi}(k-1)}{\underline{\phi}^T(k-1)\underline{\phi}(k-1) + c} + \tilde{\underline{\Theta}}^T(k)\Delta\underline{\phi}(k-1) + d(k) \quad (4.6.1.1.6)$$

Using (4.6.1.1.3) and assuming c small, the s dynamics become:

$$s(k+1) = [1 - a(k)]s(k) + \tilde{\underline{\Theta}}^T(k)\Delta\underline{\phi}(k-1) + d(k) - [1 - a(k)]d(k-1) \quad (4.6.1.1.7)$$

The adaptation gain $a(k)$ clearly has an effect on the bandwidth of these dynamics. Additionally, $\tilde{\underline{\Theta}}$ and $\Delta\underline{\phi}(k-1)$ are themselves functions of $s(k)$ which further complicates the bandwidth determination. The difficulty in accurately specifying the bandwidth of the s dynamics makes it possible for the control input to excite unmodeled dynamics.

A similar analysis can be performed if the controller uses a weighted least squares estimator, and results in the s dynamics:

$$\begin{aligned}
s(k+1) = & \left[1 - \frac{a(k)\underline{\phi}^T(k-1)\underline{P}(k-2)\underline{\phi}(k-1)}{1 + a(k)\underline{\phi}^T(k-1)\underline{P}(k-2)\underline{\phi}(k-1)} \right] s(k) + \\
& + \tilde{\underline{\Theta}}^T(k)\Delta\underline{\phi}(k-1) + d(k) - [1 - a(k)]d(k-1)
\end{aligned}
\tag{4.6.1.1.8}$$

Again, the addition of the parameter estimator is seen to greatly complicate the bandwidth control of these dynamics, and thus lead to the possibility of exciting unmodeled dynamics.

4.6.1.2 Corrupted Adaptation Signals

The presence of unmodeled dynamics can cause variations in $s(k)$ which can potentially be misinterpreted as being due to parameteric error and thus result in degraded performance of the parameter estimator. This can be seen by considering the treatment in Section 4.5.3 of the deterministic actuator dynamics in the adaptive controller input. For simplicity of discussion, the unmodeled dynamics here will be assumed to take the form of actuator dynamics, i.e.:

$$u(k) = H(z^{-1})u_c(k) \tag{4.6.1.2.1}$$

where $H(z^{-1})$ is a strictly causal and strictly stable but unknown linear pulse transfer function and $u_c(k)$ represents the commanded input. By rewriting (4.6.1.2.1) as:

$$u(k) = u_c(k) + [H(z^{-1}) - 1]u_c(k) \tag{4.6.1.2.2}$$

and interpreting the effect of the unmodeled dynamics as a disturbance to the commanded input $u_c(k)$, the total disturbance acting on the system due to the unmodeled dynamics is:

$$d(k) = \sum_{i=1}^d \left\{ \Theta_{b_i}^T \phi_{b_i}(k) z^{-(i-1)} \right\} d_u(k)$$

(4.6.1.2.3)

where:

$$d_u(k) = [H(z^{-1}) - 1]u_c(k)$$

(4.6.1.2.4)

If the disturbance is unaccounted for, it is clear that $\underline{e}^T \underline{x}(k+1)$ will be corrupted by the $d(k)$ of (4.6.1.2.3). The boundedness of the parameter estimates and convergence of the prediction error can no longer be shown, and hence the resultant adaptive controller cannot be shown to result in a bounded regressor vector or to meet the control objective.

4.6.2 Possible Solutions to Instability Mechanisms

The analysis in the last section of potential mechanisms for instability has itself implied possible solutions to overcome these mechanisms. To deal with the bandwidth modification issue, the specification of certain parameters in the adaptive controller and parameter estimator will attempt to maintain the bandwidth controlled dynamics of the scalar s , and hence of the state $\underline{x}(k)$. To correct for the corruption of adaptation signals, a method is proposed which bounds the error due to unmodeled dynamics and treats it as a disturbance in the parameter estimator as in Sections 4.4 and 4.5.

Note that one approach to dealing with these problems is to restrict adaptation to occur at a very low rate such that the closed loop system is quasi-time-invariant. The implicit approach here, however, is to maintain adaptation at the highest possible rate so as to be better suited to handle time-variant parameters and to achieve the best possible closed performance in the shortest possible time.

4.6.2.1 Bandwidth Control during the Adaptation Process

In Section 4.6.1.1, it was shown how the adaptive controller could result in a bandwidth modification to the dynamics of s and hence potentially excite unmodeled dynamics. As in Chapter 3, it is possible to modify the control law so as to achieve better control over the s dynamics. For this purpose, the control law of (4.5.2.1) is augmented with a term in $s(k)$:

$$u(k) = \frac{1}{\hat{\Theta}_{b_1}^T(k)\underline{\phi}_{b_1}(k)} \left\{ \gamma s(k) - \hat{\Theta}_f^T(k)\underline{\phi}(k) - \underline{e}^T \underline{A}' \underline{x}(k) + \sum_{i=2}^d \left\{ \hat{\Theta}_{b_i}^T(k)\underline{\phi}_{b_i} z^{-(i-1)} \right\} u(k) + \underline{e}^T \underline{x}_d(k+1) \right\} \quad (4.6.2.1.1)$$

where $0 \leq \gamma < 1$. This control law reduces to:

$$\underline{e}^T \underline{x}_d(k+1) + \gamma s(k) = \underline{e}^T \underline{A}' \underline{x}(k) + \hat{\Theta}^T(k)\underline{\phi}(k) \quad (4.6.2.1.2)$$

Subtracting (4.6.2.1.2) from the system (4.6.1) gives:

$$s(k+1) - \gamma s(k) = \tilde{\Theta}^T(k)\underline{\phi}(k) + d(k) \quad (4.6.2.1.3)$$

It is again necessary to show that the Key Technical Lemma is satisfied for this to be a viable control law. For the gradient estimator, (4.6.2.1.3) can be substituted into (4.4.1.18) to get:

$$\lim_{k \rightarrow \infty} \frac{[s(k) - \gamma s(k-1) - R(k) \text{sgn}\{g(k)\}]^2}{\underline{\phi}^T(k-1)\underline{\phi}(k-1) + c} = 0 \quad (4.6.2.1.4)$$

Conditions (1) and (2) of the lemma are satisfied as in Section 4.5.2.1. To show that Condition (3) is satisfied, it is necessary that $s(k) - \gamma s(k-1) - R(k) \text{sgn}\{g(k)\}$ bound $\underline{\phi}(k-1)$. For this purpose, introduce:

$$r(k) = s(k) - \gamma s(k-1)$$

(4.6.2.1.5)

By the choice of $0 \leq \gamma < 1$, $s(k)$ will remain bounded if $r(k)$ is bounded. Conversely, $s(k)$ will be unbounded if $r(k)$ is unbounded. Because the dynamics of (4.6.2.1.5) are linear, $r(k)$ can therefore linearly bound $s(k)$. Since $s(k)$ has already been shown to bound $\underline{x}(k)$ in Section 4.5.1.1, then $r(k)$ is also able to linearly bound $\underline{\phi}(k-d)$. Condition (3) of the lemma is also satisfied and allows for the result that $\underline{\phi}(k-1)$ is bounded and $\lim_{k \rightarrow \infty} s(k) - \gamma s(k-1) - R(k) \text{sgn}\{g(k)\} = 0$ since $R(k)$ is bounded.

As in Section 4.5.2, it is desirable to make the non-adaptive controller used inside the boundary layer compatible with the adaptive controller used outside the boundary layer. Since the desired s dynamics inside the boundary layer were characterized by the eigenvalue p , it is reasonable to also select this behavior outside the boundary layer and is accomplished by letting $\gamma = p$. $R(k)$ is again selected as in (4.5.2.1.17):

$$R(k) = (1 - p)\Phi$$

(4.6.2.1.6)

which bounds $d(k)$ as required using (4.5.2.1.11). From (4.4.1.6) and (4.6.2.1.2), $g(k)$ can be found as:

$$g(k) = s(k) - \gamma s(k-1)$$

(4.6.2.1.7)

Note that with $\gamma = p$ and the $R(k)$ of (4.6.2.1.6), (4.6.2.1.4) implies:

$$\lim_{k \rightarrow \infty} s(k) = \Phi(k) \text{sgn}\{s(k)\}$$

(4.6.2.1.8)

which shows the eventual convergence of s to the boundary layer.

If the definition of (4.6.1.1.4) is again made:

$$\underline{\phi}(k) = \underline{\phi}(k-1) + \Delta \underline{\phi}(k-1) \quad (4.6.2.1.9)$$

then the s dynamics of (4.6.2.1.3) can be written:

$$s(k+1) = \gamma s(k) + \tilde{\underline{\Theta}}^T(k) \underline{\phi}(k) + \tilde{\underline{\Theta}}^T \Delta \underline{\phi}(k) + d(k) \quad (4.6.2.1.10)$$

If the controller uses the normalized gradient parameter estimator, $\tilde{\underline{\Theta}}(k)$ can be obtained from (4.2.1.3) and substituted in (4.6.2.1.10):

$$\begin{aligned} s(k+1) = \gamma s(k) + \tilde{\underline{\Theta}}^T(k-1) \underline{\phi}(k-1) - \frac{a(k) \underline{\phi}^T(k-1) \underline{\phi}(k-1) \tilde{\underline{\Theta}}^T(k-1) \underline{\phi}(k+1)}{\underline{\phi}^T(k-1) \underline{\phi}(k-1) + c} + \\ + \tilde{\underline{\Theta}}^T(k) \underline{\phi}(k-1) + d(k) \end{aligned} \quad (4.6.2.1.11)$$

Using (4.6.2.1.10) and assuming c small, (4.6.2.1.11) becomes:

$$\begin{aligned} s(k+1) = [1 + \gamma - a(k)]s(k) + [-\gamma + \gamma a(k)]s(k-1) + \\ + \tilde{\underline{\Theta}}^T(k) \underline{\phi}(k-1) + d(k) - [1 - a(k)]d(k-1) \end{aligned} \quad (4.6.2.1.12)$$

By selecting $\gamma = p$ as discussed above and letting $a(k) = 1 - p$, the s dynamics become:

$$s(k+1) = 2ps(k) - p^2s(k-1) + \tilde{\underline{\Theta}}^T(k) \underline{\phi}(k-1) + d(k) - pd(k-1) \quad (4.6.2.1.13)$$

which mimics a critically damped, second order filter with eigenvalues (p, p) driven by parametric error and disturbances.

If the controller uses the weighted least squares parameter estimator, the corresponding expression to (4.6.2.1.12) is:

$$\begin{aligned}
 s(k+1) = & \left[1 + \gamma - \frac{a(k)\underline{\phi}^T(k-1)\underline{P}(k-2)\underline{\phi}(k-1)}{1 + a(k)\underline{\phi}^T(k-1)\underline{P}(k-2)\underline{\phi}(k-1)} \right] s(k) + \\
 & + \left[-\gamma + \frac{\gamma a(k)\underline{\phi}^T(k-1)\underline{P}(k-2)\underline{\phi}(k-1)}{1 + a(k)\underline{\phi}^T(k-1)\underline{P}(k-2)\underline{\phi}(k-1)} \right] s(k-1) + \\
 & + \tilde{\underline{\Theta}}^T(k)\underline{\phi}(k-1) + d(k) - [1 - a(k)]d(k-1)
 \end{aligned}
 \tag{4.6.2.1.14}$$

To have these dynamics mimic the second order filter structure of (4.6.2.1.13), it is necessary to make $a(k)$ time-variant due to the time-variant nature of $\underline{P}(k)$:

$$a(k) = \frac{1 - p}{p\underline{\phi}^T(k-1)\underline{P}(k-2)\underline{\phi}(k-1) + c}
 \tag{4.6.2.1.15}$$

where c is a small positive constant that keeps $a(k)$ bounded when $\underline{\phi}^T(k-1)\underline{P}(k-2)\underline{\phi}(k-1) = 0$. When $\underline{P}(k)$ is large at the initiation of the controller action, $a(k)$ must be kept small to limit excessive parameter variations that could excite unmodeled dynamics. Similarly, when $\underline{P}(k)$ becomes small with time, it is necessary to increase $a(k)$ such that adaptation occurs at a reasonable rate.

The advantage of using the augmented control law versus the original version is that the dynamics of s become controllable at higher (second) order than is possible without the augmentation. This allows for a sharper frequency roll-off in the filter to reduce the possibility of higher frequency content being present in s that could excite unmodeled dynamics.

4.6.2.2 Relevant Adaptation

From the discussion in Section 4.6.1.2, it is clear that unmodeled dynamics can have an effect similar to a disturbance if they are assumed to take the form of actuator dynamics. The approach taken in Section 4.5.3 for deterministic actuator dynamics could be used if the pulse transfer function $H(z^{-1})$ was known. In the general case, of course, this function is not known and thus a means must be devised for estimating or bounding its effect.

Before investigating methods for accounting for the unmodeled dynamics, it is important to examine the frequency range that must be considered in the analysis. In the digital implementation of the control law, its frequency content is substantially limited to the Nyquist frequency. There is therefore little reason to consider unmodeled dynamics more than about five times this frequency as the control input will be unable to contribute significant energy above this frequency.

From Section 4.6.1.2, the additional disturbance to be considered due to unmodeled dynamics is of the form:

$$d(k) = \sum_{i=1}^d \left\{ \Theta_{b_i}^T \phi_{b_i}(k) z^{-(i-1)} \right\} d_u(k) \quad (4.6.2.2.1)$$

where:

$$d_u(k) = [H(z^{-1}) - 1]u_c(k) \quad (4.6.2.2.2)$$

The disturbance of (4.6.2.2.1) can be bounded:

$$D(k) \geq \left| \sum_{i=1}^d \left\{ \Theta_{b_i}^T \phi_{b_i}(k-1) z^{-(i-d-1)} \right\} d_u(k) \right| \quad (4.6.2.2.3)$$

Let the bound on $d_u(k)$ be given by:

$$|d_u(k)| = \max_{0 < w < w_u} \left\{ | [H(z^{-1}) - 1]u_c(k) | \right\} \quad (4.6.2.2.4)$$

where w represents the frequency range of interest, which extends to the upper frequency limit w_u (five times the Nyquist frequency) as discussed above.

The maximum limit on $d_u(k)$ cannot in general be computed due to the uncertainty on the unmodeled dynamics $H(z^{-1})$. Several methods can be used to approximate this bound in the relevant frequency range if a reasonable structure for $H(z^{-1})$ can be deduced by some knowledge of the unmodeled dynamics. These methods normally assume that the unmodeled dynamics have a linear structure and that a frequency range of interest can be specified.

One approach is to consider the function $H(z^{-1})$ and input $u_c(k)$ in the frequency domain which obtains a bound on the magnitude response of $H(jw)$ (the continuous time counterpart to $H(z^{-1})$) over the entire frequency range of interest. A discrete fourier transform is performed on $u_c(k)$ (using a small number of samples) to determine its magnitude at a particular frequency. These two magnitudes are multiplied together to obtain a bound on the error on the unmodeled dynamics [54]. The frequency response of $H(jw)$ is determined through *a priori* analysis in which its magnitude is evaluated for a finite number of difference frequencies within the frequency spectrum, and the greatest magnitude found at each frequency found by searching through a finite number of possible pole/zero combinations that suitably cover the relevant frequency region. The discrete fourier transform is performed using a finite sampling of the input $u_c(k)$ into the past to attempt and determine the present frequency content of $u_c(k)$. The necessity of implementing this procedure in real time limits the number of frequencies that can be considered and thus limits its accuracy. There is also some inaccuracy introduced by using the frequency domain bounds for $H(jw)$ in what is essentially a transient input signal $u_c(k)$.

The method proposed here takes place entirely in the time domain and thus is well suited to a real time control application. This method can be summarized as follows:

1. Determine a simple continuous time structure for the unmodeled dynamics that

are of interest in the relevant frequency range (e.g. a first or second order linear filter) of the physical system being considered.

2. Select a finite number of continuous time parameters that characterize the unmodeled dynamic structure of (1) that bound the extremities. In a second order filter structure, for example, this would be done by considering a range of natural frequencies and damping ratios that include as a member the unmodeled dynamics present.
3. Discretize each filter (i.e. each filter is associated with a certain damping ratio and natural frequency) to obtain the corresponding expression in discrete time.
4. Set up a parallel filter arrangement composed of the individual filters of (3) with common input $u_c(k)$. The maximum absolute value of all the filter outputs is used to bound $|d_u(k)|$. Formally:

$$|d_u(k)| = \max_j \{ | [H_j(z^{-1}) - 1]u_c(k) | \} \quad (4.6.2.2.5)$$

where j represents each individual filter of (3).

5. The expression of (4.6.2.2.3) is then used to bound the disturbance $d(k)$ in the system description. If Θ_{b_i} is unknown, it is further necessary to bound $\hat{\Theta}_{b_i}^T(k)$.
6. This time-variant disturbance bound is then used along with the methods of Section 4.4 in the adaptive controller design which attempts to preserve the system stability by maintaining the properties of the parameter estimator (i.e. adaptation using relevant signals only).

This method has the advantage that, as the number of filter used (j) approaches infinity, the $|d_u(k)|$ of (4.6.2.2.5) will be guaranteed to bound the actual error due to the (unknown) function $H(z^{-1})$. Furthermore, if the unmodeled dynamics are known

exactly, this method converges to that of Section 4.5.3 where only a single filter is necessary.

2. Select a finite number of continuous time parameters that characterize the unmodeled dynamic structure of (1) that bound the extremities. In a second order filter structure, for example, this would be done by calculating a range of natural frequencies and damping ratios that include as a member the unmodeled dynamics present.

3. Distribute each filter (i.e. each filter is associated with a certain damping ratio and natural frequency) to obtain the corresponding expression in discrete time.

4. Set up a parallel filter arrangement composed of the individual filters of (3) with common input $w(k)$. The maximum absolute value of all the filter outputs is used to bound $\hat{d}_i(k)$. Formally:

$$\hat{d}_i(k) = \max \left\{ \left| \hat{H}_i(e^{-j\omega}) - \hat{H}_i(k) \right| \right\}$$

(4.5.3.2)

where \hat{H}_i represents each individual filter of (3).

5. The expression of (4.5.3.2) is then used to bound the disturbance $\hat{d}_i(k)$ in the system description. If \hat{d}_i is unknown, it is further necessary to bound $\hat{d}_i(k)$.

6. This time-varying disturbance bound is then used along with the methods of Section 4.2 in the adaptive controller design which attempts to preserve the system stability by maintaining the properties of the parameter estimator (i.e. adaptive gain being relevant again only).

This method has the advantage that, as the number of filter used (i) approaches infinity, the $\hat{d}_i(k)$ of (4.5.3.2) will be guaranteed to bound the actual error due to the (unknown) function $\hat{H}(e^{-j\omega})$. Furthermore, if the unmodeled dynamics are known

Example 4.3

The system of (Ex. 4.2.1) is again considered:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.22 \\ 0 & 0.95 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{0.0066}{1+0.05|x_2(k)|} \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d(k) \quad (\text{Ex. 4.3.1})$$

where $|d(k)| \leq 0.2$. The available model is again given by (Ex. 4.2.2):

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.22 \\ 0 & f' \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{b'}{1+0.05|x_2(k)|} \end{bmatrix} u(k) \quad (\text{Ex. 4.3.2})$$

with $0.65 \leq f' \leq 0.95$ and $0.0066 \leq b' \leq 0.0103$. The parameter and regressor vectors are scaled as in Example 4.2:

$$\underline{\phi}(k) = [x_2(k) \quad \frac{u(k)}{100\{1 + 0.05 |x_2(k)|\}}] \quad (\text{Ex. 4.3.3})$$

$$\hat{\underline{\theta}}(k=0) = [0.8 \quad 0.82]^T \quad (\text{Ex. 4.3.4})$$

This example will examine the effect of unmodeled dynamics by passing the control law commanded by the controller through a stable, linear, second order filter with pulse transfer function $H(z^{-1})$:

$$u(z) = H(z^{-1})u_c(z) \quad (\text{Ex. 4.3.5})$$

where $u_c(z)$ is the commanded control input. Let $H(z^{-1})$ be of the form:

$$H(z^{-1}) = \frac{0.1857z^{-1} + 0.1616}{1 - 1.325z^{-1} + 0.6724z^{-2}} \quad (\text{Ex. 4.3.6})$$

which has poles $0.6625 \pm 0.4832i$.

The control laws of (Ex. 4.2.5) - (Ex. 4.2.8) are used depending on whether s is inside or outside the boundary layer. The normalized gradient parameter estimator with bounded disturbances of (Ex. 4.2.9) - (Ex. 4.2.10) is also used, with $a(k) = 1$.

As pointed out in Section 4.6.1, the presence of the unmodeled dynamics can invalidate the previous stability arguments that were based on the system without unmodeled dynamics. Specifically, s is not bandwidth limited outside the boundary layer and the unmodeled dynamics can cause excursions in s outside the boundary layer leading to faulty parameter estimator properties.

A simulation was performed using the desired trajectory for $x_{2_d}(k)$:

$$x_{2_d}(k) = 10\sin(0.033k) - 10\sin(0.066k) \quad (\text{Ex. 4.3.7})$$

with trajectory consistency requirement (Ex. 2.2.4) and disturbance:

$$d(k) = 0.2\sin(\pi/2) \quad (\text{Ex. 4.3.8})$$

The resultant s trajectory can be seen in Figure 4.8, where instability is seen to set in by approximately 40 sampling periods. Similarly, the parameter error norm shown in Figure 4.9 does not decrease as in previous examples. The use of the standard algorithms are therefore unsuitable in the presence of the unmodeled dynamics.

The methods of Section 4.6.2 can be used in an attempt to allow the parameter adaptation process to proceed in a stable manner. In order to provide bandwidth

control during the adaptation process, the augmented control law of (4.6.2.1.1) is used when outside the boundary layer:

$$u(k) = \frac{1}{\hat{\Theta}_{b_i}^T \Phi_{b_i}(k)} \left\{ \gamma s(k) - \hat{\Theta}_f^T(k) \phi_f(k) - \underline{e}^T A' \underline{x}(k) + \right. \\ \left. - \sum_{i=2}^d \left\{ \hat{\Theta}_{b_i}^T(k) \phi_{b_i}(k) z^{(i-1)} \right\} u(k) + \underline{e}^T \underline{x}_d(k+1) \right\} \quad (\text{Ex. 4.3.9})$$

where $\gamma = p = 0.89$ to match the dynamics on the surface. The parameter estimator of (4.4.1.4) is used:

$$\hat{\Theta}(k) = \hat{\Theta}(k-1) + \frac{a(k) \phi(k-1) g_{\Delta}(k)}{\phi^T(k-1) \phi(k-1) + c} \quad (\text{Ex. 4.3.10})$$

where $a(k) = 1 - p = 0.11$ as suggested in Section (4.6.2.1), and:

$$g_{\Delta}(k) = \begin{cases} s(k) - \gamma s(k-1) - (1-p)\Phi \text{sgn}\{s(k)\} & \text{if } |s(k)| > \Phi \\ 0 & \text{otherwise} \end{cases} \quad (\text{Ex. 4.3.11})$$

To ensure that adaptation is only performed when relevant information is available, the techniques of Section 4.6.2.2 are used to bound the effect of the unmodeled dynamics on s . This is done by utilizing the disturbance term due to unmodeled dynamics from (4.6.2.2.2):

$$d_u(k) = [H(z^{-1}) - 1]u_c(k) \quad (\text{Ex. 4.3.12})$$

and augmenting the disturbance bound D with $d_u(k)$ to produce a time-variant boundary layer $\Phi(k)$. In this example, the unmodeled dynamics were treated in a deterministic way - that is, $H(z^{-1})$ was assumed known. The treatment of the more general case

of unknown unmodeled dynamics will be demonstrated in the experimental analysis in Chapter 6.

The performance of the modified algorithm can be seen in Figure 4.10. The s trajectory exhibits a bandwidth limited behavior outside the boundary layer as well as inside the layer. By comparison with Figure 4.6 of Example 4.2, the boundary layer is larger due to the increased disturbance term $d_u(k)$ which arises from the unmodeled dynamics. Figure 4.11 illustrates the control input resulting from the modified algorithm. Any high frequency activity in the control input is seen to increase the size of the boundary layer due to the bounding of the unmodeled dynamics - this can be observed between sample numbers 100 and 120.

The parameter estimator performance is summarized by the parameter error norm of Figure 4.1 which is non-increasing. The individual parameter estimates are given in Figures 4.13 and 4.14. The estimate for parameter 1 approaches its actual value but the estimate for parameter 2 is seen to change very little. This again is a consequence of the use of the boundary layer to ensure stability and maintain the convergent properties of the parameter estimator.

A key issue in this example has been whether the value $p = 0.89$ was suitably chosen for the unmodeled dynamics present. This topic will be covered in the next chapter when bandwidth selection is discussed.

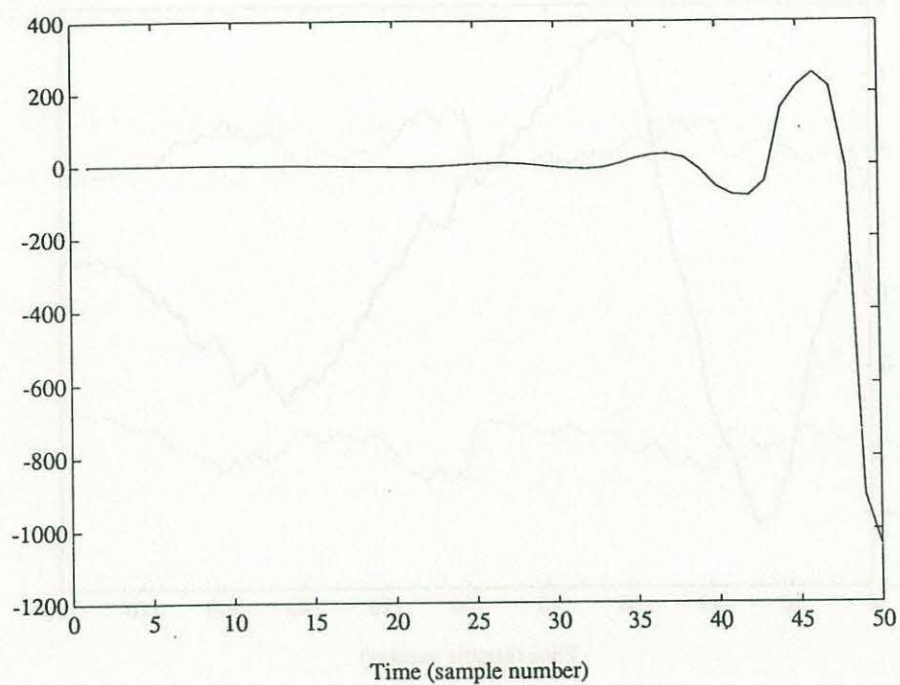


Figure 4.8 S trajectory

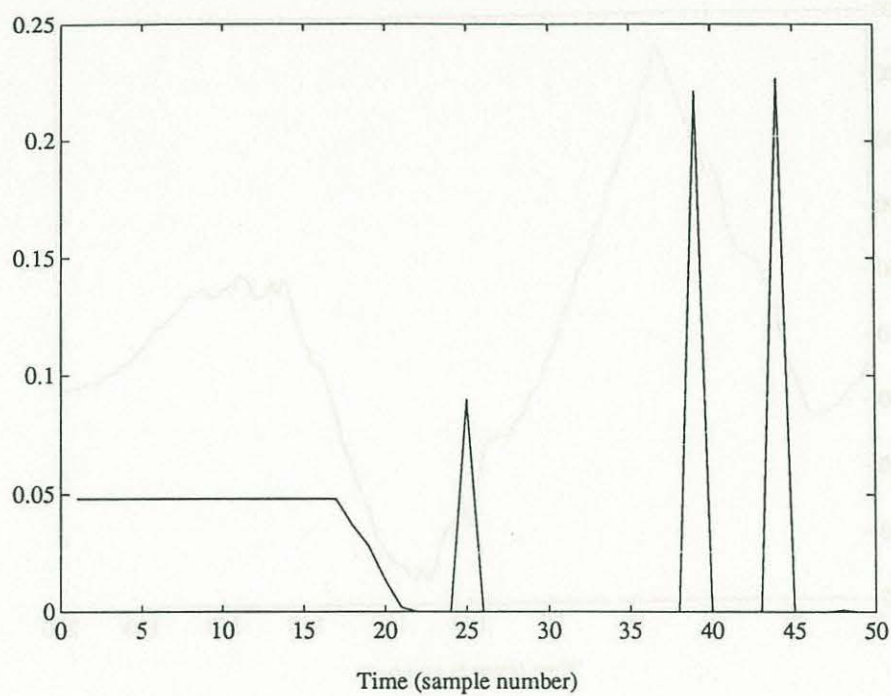


Figure 4.9 Parameter error norm

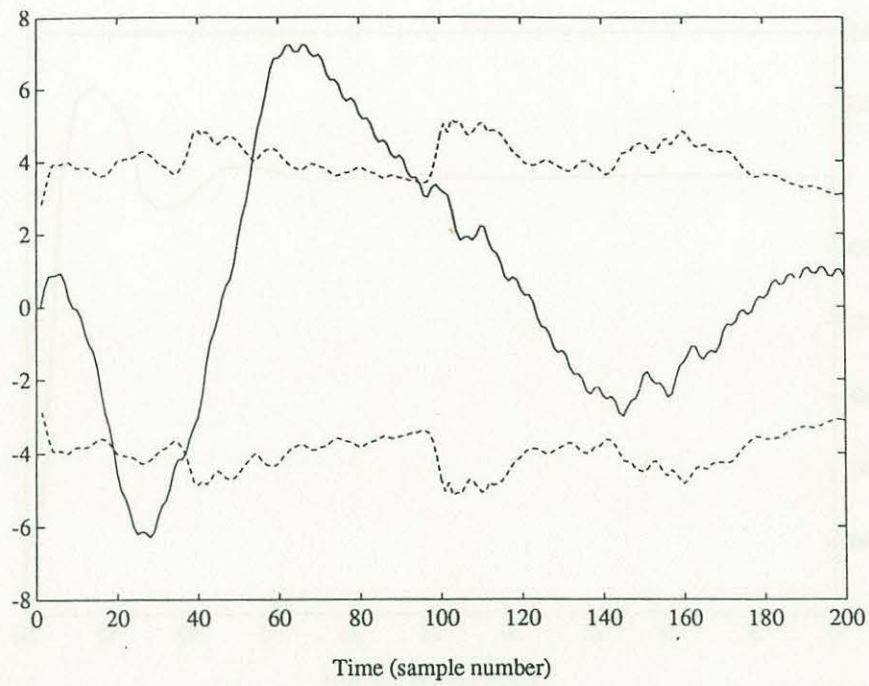


Figure 4.10 S (-) and Φ (- -)

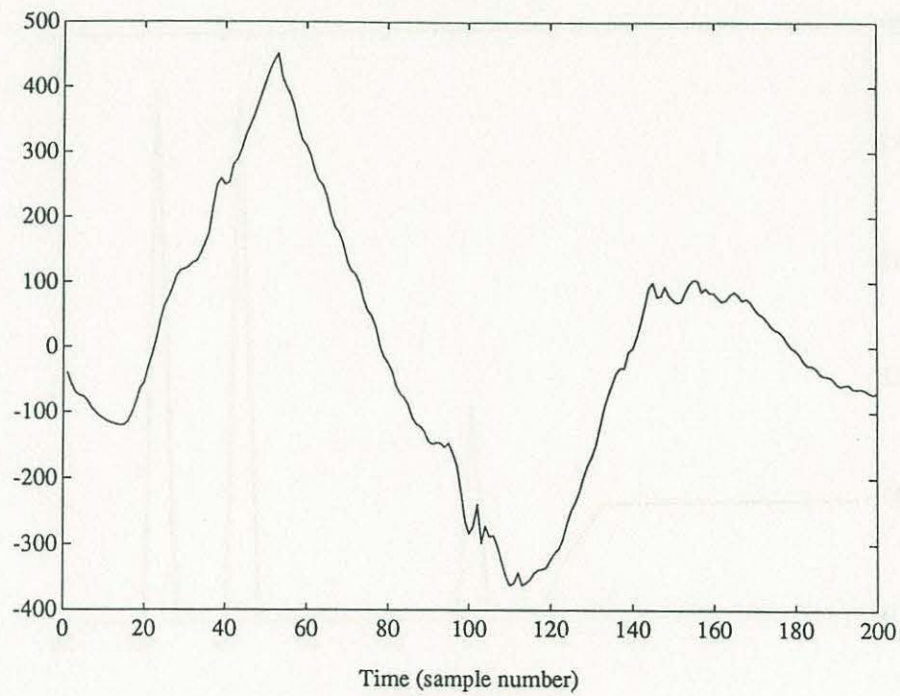


Figure 4.11 Control input

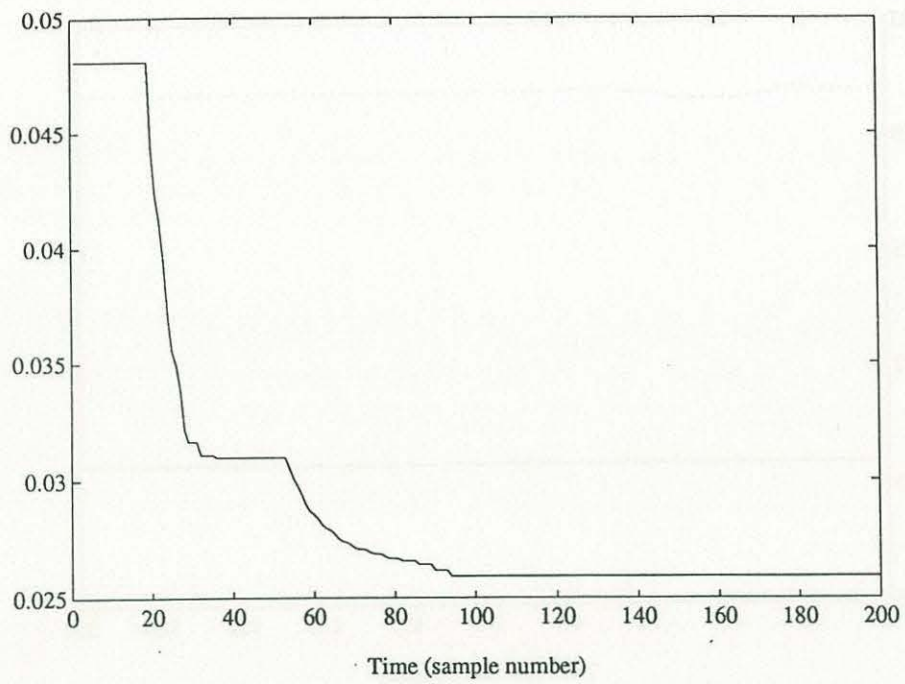


Figure 4.12 Parameter error norm

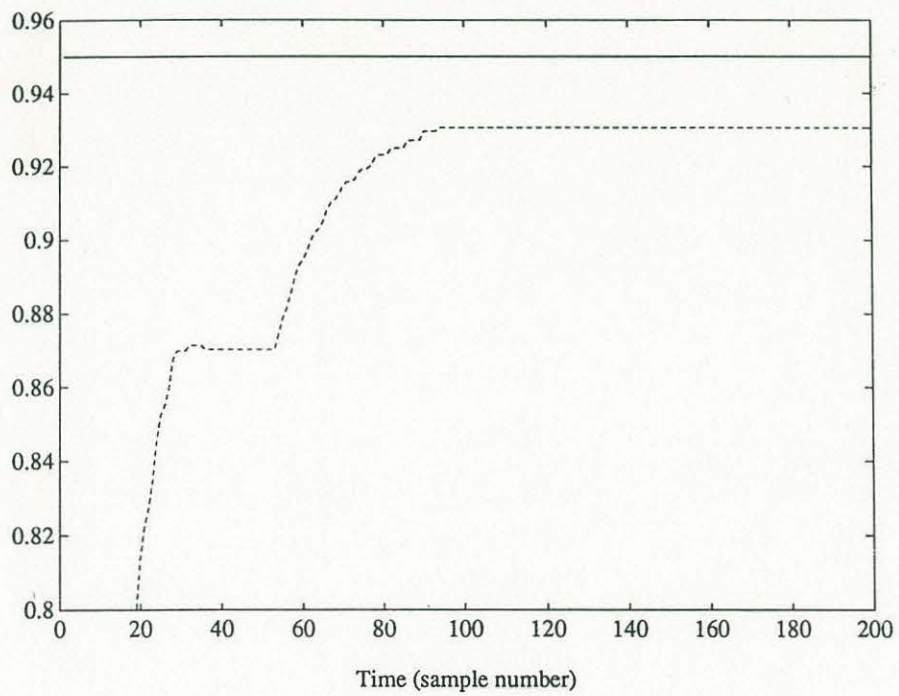


Figure 4.13 Actual (-) and estimated (- -) parameter 1

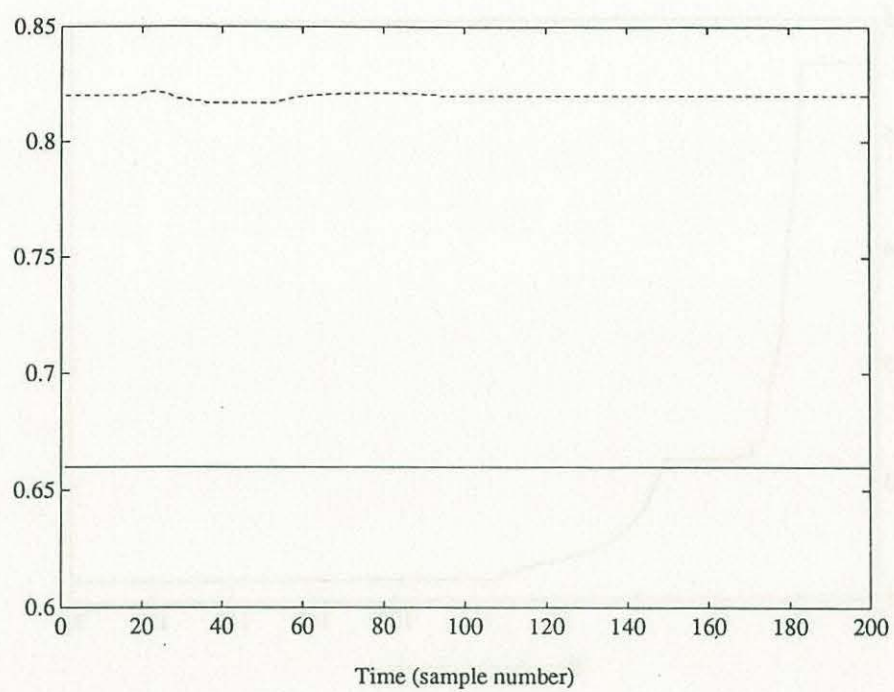


Figure 4.14 Actual (-) and estimated (- -) parameter 2

Chapter 5

Physical System Implementation

The preceding chapters have concentrated on the theoretical aspects of controlling a class of discrete time dynamic systems. In this chapter the emphasis will be directed toward applying these control techniques to physical systems. The term “physical system” is used here to refer to systems with continuous time dynamics.

5.1 Modeling

5.1.1 System Identification

A requisite first step in applying any control technique is the acquisition of a system model upon which to base the design. The system identification process is composed of two major steps - determination of the model structure and determination of a parameter set that characterizes that particular model structure. Elements that determine the structure of the model include the order (number of state variables), linear or nonlinear combinations of the state variables, how disturbances and deterministic inputs enter the system, and the pure time delays present. The parameter set that characterizes this model structure is normally comprised of constant elements which depend on the scale of the system under consideration.

Frequently the model structure can be deduced using laws from physical dynamics

such as Newton's Law, etc. In this case a model structure and parameter set can be developed analytically with high accuracy. In less understood areas such as hydrodynamics, it is much more difficult to formulate well structured and relatively low order dynamic models because of the uncertainty of the physical process involved. In these cases, an empirical modeling procedure is frequently used to formulate a model with a relatively simple structure characterized by a constant parameter set. The field of system identification has developed as an empirical modeling technique that generally uses discrete input-output information about a system coupled with parameter estimation methods to formulate a discrete time model [78, 79]. These techniques have evolved in a discrete time manner due to the nature of the hardware used to collect the data – frequently digital computer systems which collect data at discrete time intervals.

If the system model is being determined for the purpose of control system design, it is next necessary to ascertain the compatibility of the proposed model with the control design method to be used. For example, if a discrete time method is used the model may be able to be used directly (or with little modification) due to the discrete time nature of the model. If a continuous time method is used, however, the model must first be converted to its continuous time equivalent. Unfortunately, this conversion may not be possible to accomplish exactly, especially if the system identification method was used because of the lack of detailed knowledge available about the physical principles guiding the system dynamics or if significant nonlinearities or time delays are present. Any approximation at this step will have some effect on the effectiveness or predictability of the design method, and will be further complicated if the already approximated method must itself be implemented (as described in Chapter 1) in a discrete time manner. This issue will be discussed more fully in Section 5.2.

5.1.2 Discrete Time Modeling of Continuous Time Systems

Somewhat contrasting with the previous section is the situation where a discrete time control design method is to be used with a physical system with some known

structural form. In this case it is necessary to find a corresponding discrete time model to the continuous time, analytically derived model.

The most common method of calculating and commanding inputs in a control system using a digital computer normally implements the control law in a piecewise constant fashion using a zero-order-hold device located in a digital-to-analog converter. If the resulting continuous time system dynamics are expressed in linear, time-invariant form, they can be integrated exactly with a piecewise constant control input to yield an exact discrete time system expression [70, 71]. Nonlinear, time-variant dynamics, however, cannot in general be solved exactly and thus do not possess an exact discrete time analog. This requires the use of a methodology to discretize these types of systems.

The procedure used here approximates the continuous time system by a piecewise linear, time-variant discrete time system. Using concepts presented in [80], this procedure is based on the integration of a piecewise linear, constant coefficient differential equation whose coefficients are the Jacobian of the nonlinear system and whose final state (after a piecewise constant control input) is close to that of the nonlinear system. This method was used because it produces a model which is explicitly linear in the control input when the continuous time system is also linear in the control input (a characteristic not shared by some other integration schemes like, for example, the Runge-Kutta method) which is required to support the development of the preceding chapters.

Let the nonlinear, time-variant, continuous time single-input system be:

$$\dot{\underline{x}} = \underline{f}(\underline{x}, u, t) \quad (5.1.2.1)$$

Differentiate (5.1.2.1) with respect to time:

$$\ddot{\underline{x}} = \frac{\partial \underline{f}}{\partial \underline{x}} \dot{\underline{x}} + \frac{\partial \underline{f}}{\partial u} \dot{u} + \frac{\partial \underline{f}}{\partial t} \quad (5.1.2.2)$$

Now consider (5.1.2.2) over a discrete time interval. The Jacobian is by definition $\underline{J} = \partial \underline{f} / \partial \underline{x}$, and the zero-order-hold nature of the digital-to-analog converter makes $\dot{u} = 0$ over the sampling interval. If $(\partial \underline{f} / \partial t)_o$ and \underline{J}_o are evaluated at the start of the interval (as indicated by the "o" subscript) and held constant over the time interval, the (5.1.2.2) becomes a piecewise, time-variant differential equation with a piecewise constant input.

The homogeneous equation to be solved is:

$$\ddot{\underline{x}} - \underline{J}_o \dot{\underline{x}} = 0 \quad (5.1.2.3)$$

which has the solution for time interval $t = 0$ to $t = T$:

$$\dot{\underline{x}} = \underline{e}^{\underline{J}_o T} \dot{\underline{x}}_o \quad (5.1.2.4)$$

where $\dot{\underline{x}}_o = \dot{\underline{x}}(t = 0)$. The complete solution for $\partial \underline{f} / \partial t$ piecewise constant (note $\partial \underline{f} / \partial t = 0$ for a time-invariant system) is:

$$\dot{\underline{x}} = \underline{e}^{\underline{J}_o T} \dot{\underline{x}}_o + \int_0^T \underline{e}^{\underline{J}_o \tau} (\partial \underline{f} / \partial t)_o d\tau \quad (5.1.2.5)$$

which is integrated again to yield:

$$\underline{x} = \underline{x}_o + \int_0^T \underline{e}^{\underline{J}_o \tau} \dot{\underline{x}}_o d\tau + \int_0^T \int_0^T \{ \underline{e}^{\underline{J}_o \tau} (\partial \underline{f} / \partial t)_o \} d\tau dt \quad (5.1.2.6)$$

where $\underline{x}_o = \underline{x}(t = 0)$. This can be expressed in discrete time form as:

$$\underline{x}(k+1) = \underline{x}(k) + \int_0^T e^{\underline{J}_k \tau} \underline{\dot{x}}_0 d\tau + \int_0^T \int_0^T \left\{ e^{\underline{J}_k \tau} (\partial \underline{f} / \partial t)_k \right\} d\tau dt \quad (5.1.2.7)$$

where the subscript "k" indicates the quantity is evaluated at time $t = kT$. Note that the equilibrium of (5.1.2.1) corresponds to $\underline{\dot{x}} = \underline{f}(\underline{x}, u, t) = \frac{\partial \underline{f}}{\partial t} = 0$ which is also the equilibrium point of (5.1.2.7). This transformation from the continuous time system to the resulting discrete time model involves two approximations which occur from integrating the Jacobian (which is evaluated once at the beginning of the sampling interval) over the sampling interval and from assuming that $\partial \underline{f} / \partial t$ is constant over the sampling interval. This procedure can also be used to map continuous time model uncertainty into discrete time model uncertainty.

Because this method involves some approximation, the actual closed loop system will not behave exactly as predicted since an approximate model has been used in the control design. The implications of this on the resultant system will be addressed in the next section.

5.2 Continuous Time Versus Discrete Time Controller Formulation

The control practitioner must initially decide on the use of a continuous or discrete time approach for proceeding with the control design. This is actually a subtle choice that can be influenced by many factors.

Because the physical system actually possesses continuous time dynamics and can often be modeled with a continuous time model, a continuous time design method would seem to be most appropriate for systems of this type. There is also a considerable amount of engineering insight which can often be applied to continuous time based controllers that can facilitate the design and implementation cycle. However, the sensor inputs and the control law implementation are commonly done in a discrete time manner which introduces an approximation into the resulting closed loop system which is difficult to quantify. This approximation can be made arbitrarily small by increasing the sampling rate to a high value. This assumes that adequate computational power is available and that sensor inputs can be obtained at a similarly high rate. If constraints do exist in these areas, however, it is often possible to limit the desired closed loop bandwidth such that the approximation induced by sampling becomes relatively small. This limitation is often expressed as a "thumb rule" which requires that, when characterizing the closed loop system as having a first order type system behavior, the rise time be at least five sampling intervals long [71].

One interpretation of sampling rate limitations of this form is that of ensuring a stable (and predictable) mapping from the continuous time to discrete time domain. When exceeding this limitation, the resulting closed loop system will not necessarily be unstable but will have a different closed loop bandwidth than the design bandwidth. It can be argued that with a more in depth analysis, the continuous time control law could be designed such that in the resulting discrete time implementation the desired closed loop characteristics would be achieved. However, this is exactly what motivates

the use of a discrete time control methodology in that the discrete time implementation is explicitly accounted for throughout.

The discrete time controller formulation depends upon the availability of a discrete time system model on which the design is based. As discussed in Section 5.1, this model sometimes arises directly from a discrete time system identification procedure or from the discretization of a continuous time model assuming a zero-order-hold on the input. From Section 5.1.2., however, this procedure can result in an approximate (rather than exact) discrete time model to base the controller on, and therefore introduce errors which could result in closed loop performance different than that which was intended.

An advantage of the discrete time formulation is the ability to deal with time delays in the control input (which are integral multiples of the sampling period) in a reasonably straightforward manner, whereas this is not possible in the continuous time formulation. A disadvantage of the discrete time approach, however, is the loss of physical insight that occurs after discretization has been performed.

The final decision over which controller type to use will inevitably involve some trade-off of the advantages and disadvantages list here. For the case of a nonlinear, continuous time system of known structure, a discretization error will occur with the continuous time controller in the implementation stage. This implementation error can be eliminated with the use of a discrete time controller, but in this case a similar error appears in the process of obtaining a discrete time model. The continuous time controller is probably preferable in this case because of the insight it offers into the physical system. If the continuous time controller bandwidth is being limited by the sampling rate (see Section 5.4.) in order to merely produce a predictable map into the discrete time domain during digital implementation, however, then a discrete time formulation should be investigated. If the physical system structure is somewhat vague and a discrete time model is produced directly (e.g. via system identification methods), then the best choice is probably to apply the discrete time formulation directly rather than approximate the discrete time model by a continuous time model and then imple-

menting a continuous time controller (based on this approximate model) in a discrete time manner (introducing yet another approximation).

5.3 System Form

Related to the previous section's discussion concerning the use of continuous time or discrete time formulated controllers is the issue of system form, or structure which appears as a requirement in earlier chapters. Specifically, the development thus far has required that the system model be in the form of (3.1) so that the (possibly unknown) nonlinear and time-variant effects were present in a single dimension only. A key question concerns the procedure to be followed if the system is not in this form.

First consider the case where the system is completely known but has nonlinearities and/or time-variant effects present in more than one dimension. One possible approach is to attempt to find a nonlinear transformation that can place the system in the form of (3.1). This is in fact a current research topic in which the use of Lie algebra has been employed to find such transformations ([13, 81-85]).

A much more difficult problem exists when there are unknown elements in the original system (whether linear and time-invariant or nonlinear and time-variant) which are not confined to one dimension. Admittedly, a transformation may exist which can convert the system form to the required type and retain some representation of the uncertainty in the new state variables. However, that same transformation must be applied to the original state vector to obtain the new state vector in the transformed space. Since that transformation itself is uncertain, the new state vector will also contain uncertainty. This is significant since the controller structures used here depend on full state feedback being available, and implicit in this is that the state variables being used are correct. Although the resultant uncertainty in the state vector could conceivably be accounted for in the controller formulation, this would be very difficult to develop (unless, of course, the uncertainty is known *a priori* to be bounded in which case it can be treated as a bounded disturbance).

Another issue regarding system form involves the linear boundedness assumptions of Chapters 3 and 4. Practically speaking, this limits the direct applicability of the method to linear systems which are open loop stable or unstable, and to nonlinear

systems which are open loop stable or unstable (but linearly boundable). This does leave the class of nonlinear systems which are unstable at a higher than linear rate between sampling intervals, but systems with these characteristics present a formidable challenge to any control methodology.

5.4 Bandwidth Issues

A bandwidth limited control law has been pursued in Chapters 3 and 4 in order to minimize the excitation of the (inevitable) unmodeled dynamics present in a physical system. It is important to now consider how the control bandwidth should be selected based on some knowledge of the neglected dynamics.

As this methodology is principally designed for use with continuous time physical systems, the bandwidth selection will be done using constraints that originate in continuous time. A summary of constraints for the continuous time sliding mode formulation can be found in Asada and Slotine [86] for robotics applications where the bandwidth represented as λ is typically limited by three factors:

- (i) Structural Resonant Modes

$$\lambda < \frac{2\pi}{3}\nu_r \quad (5.4.1)$$

where ν_r is the structural resonance in cycles/time.

- (ii) Neglected Time Delays

$$\lambda < \frac{1}{3T_d} \quad (5.4.2)$$

where T_d is the largest unmodeled time delay.

- (iii) Sampling Rate

$$\lambda < \frac{\nu_s}{5} \quad (5.4.3)$$

where ν_s is the sampling frequency.

Constraints (i) and (ii) apply fairly directly in the discrete time case as well. Because these rules result in continuous time bandwidth constraints, however, it is necessary

to find the corresponding discrete time constraint. As this bandwidth constraint is applied directly to the linearized s dynamics in continuous time, it is reasonable to have discrete time s dynamics follow a sample version of the continuous time behavior. This can be accomplished by the simple mapping:

$$p = e^{-\lambda T} \quad (5.4.4)$$

where p is the discrete time eigenvalue that characterizes the discrete time s dynamics, λ is the applicable continuous time bandwidth constraint, and T the sampling interval.

Constraint (iii) is not directly applicable to the discrete time case since this constraint arises from the desirability of maintaining a predictable map from the continuous time formulation to the discrete time (zero-order-hold) implementation as explained in Section 5.2. Because the discrete time formulation explicitly accounts for the sampled nature of the sensor inputs and control output in its derivation, no similar limitation exists. It is, however, necessary to consider that the desired trajectory $\underline{x}_d(k)$ will probably be specified in continuous time, and thus the sampling rate must be fast enough to adequately represent the desired trajectory using its simple zero-order-hold reconstruction. As always in a sampled data system, the sampling rate should always be fast enough such that the open loop behavior is observable (i.e. no hidden oscillations exist). The desired trajectory should itself be bandwidth limited to avoid excitation of the unmodeled dynamics as well.

The preceding arguments for the selection of a desired bandwidth have been on a heuristic basis only. It is, of course, possible to determine the appropriate limit based on a more rigorous foundation. This can be most simply done by considering the scalar s dynamics as representative of the multi-dimensional, actual dynamics. Figure 5.1 shows a block diagram of the s dynamics in the frequency domain where the input consists of modeling errors and disturbances, $G(z)$ is the open loop transfer function, and the p is given by (5.4.4). For the case of no unmodeled dynamics, $G(z)$ is the unit

delay z^{-1} , and the single pole of the resultant closed loop system is simply equal to p . With unmodeled dynamics (of some known frequency range) included in $G(z)$, it is possible to determine stability limits on p using standard single input-single output methods (e.g. root locus). This analysis can also be conducted in the time domain using methods similar to those in Chapter 3.

An interesting concept for bandwidth control that can be explored in the discrete time context is that of sampling rate manipulation. As discussed in [63], it is possible to increase robustness to unmodeled dynamics in some circumstances by reducing the sampling rate. This occurs because the frequency content able to be introduced into the system by the control input decreases as the sampling rate decreases. The reduction in sampling rate must be balanced by the increase in susceptibility to disturbances which occurs, however.

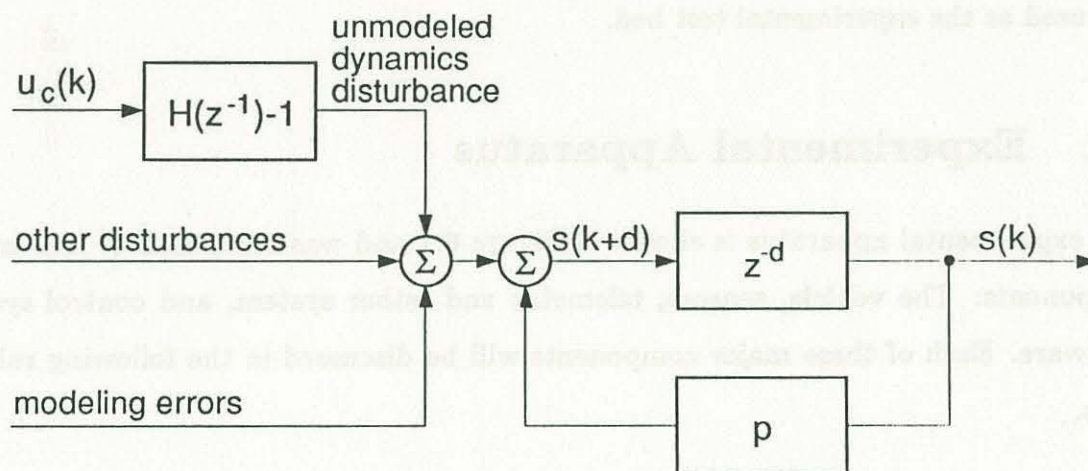


Figure 5.1 Block diagram of s dynamics

Chapter 6

Experimental Study

The results of applying the concepts presented to a physical system will be presented in this chapter to evaluate the effectiveness of the methodology in a practical environment. As detailed in the first chapter, a remotely operated underwater vehicle was used as the experimental test bed.

6.1 Experimental Apparatus

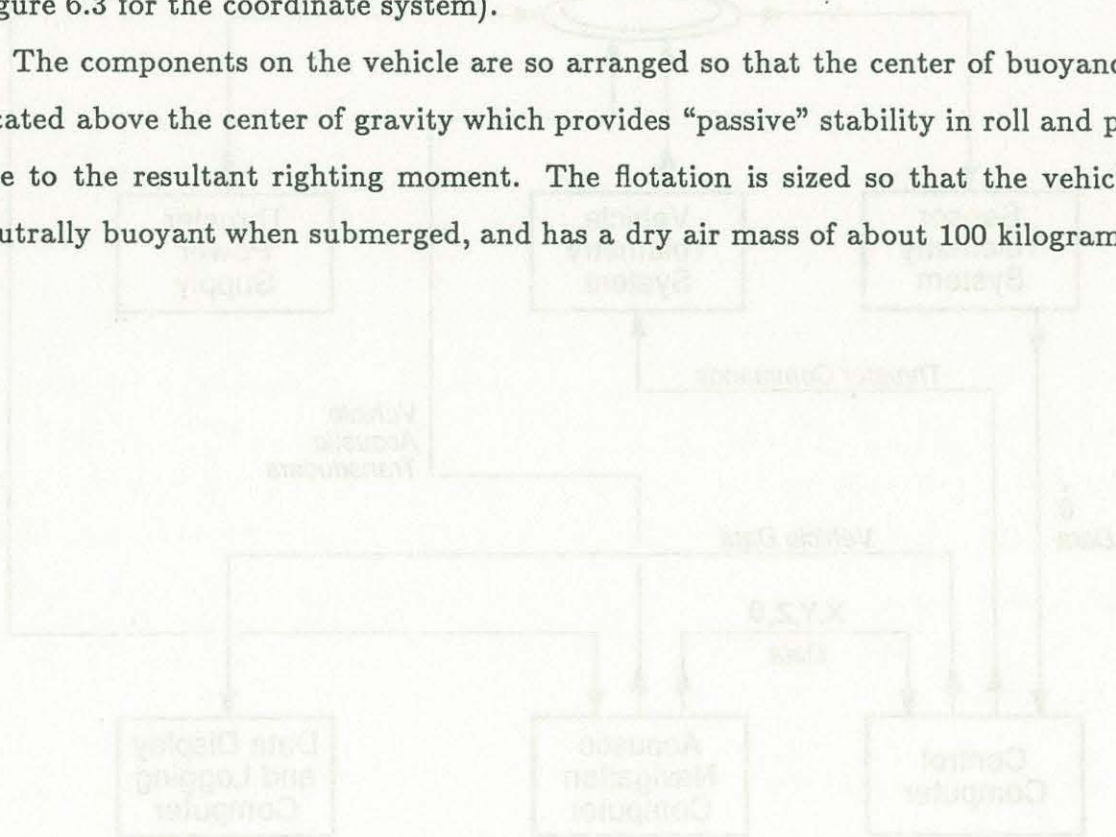
The experimental apparatus is shown in Figure 6.1 and was composed of four major components: The vehicle, sensors, telemetry and tether system, and control system hardware. Each of these major components will be discussed in the following subsections.

6.1.1 Vehicle Description

The experiments were performed using the "Jason Junior" (JJ) underwater vehicle which was built as a prototype for concepts to be used on the "Jason" vehicle (see Section 1.1). Figure 6.2 shows the general construction of JJ. A central titanium one atmosphere pressure vessel contains the telemetry system decoding electronics and

Four thrusters, each consisting of a brushless direct current motor combined with a ducted two blade propeller, provide for vehicle mobility. Each thruster is capable of delivering about 100 newtons of thrust, and has an internal controller that operates in a torque control mode (the torque applied by the motor is proportional to the thrust resulting from propeller rotation). The four thrusters are so arranged on the vehicle such that any desired motion along the x, y, z , or Θ axes could be accomplished (see Figure 6.3 for the coordinate system).

The components on the vehicle are so arranged so that the center of buoyancy is located above the center of gravity which provides "passive" stability in roll and pitch due to the resultant righting moment. The flotation is sized so that the vehicle is neutrally buoyant when submerged, and has a dry air mass of about 100 kilograms.



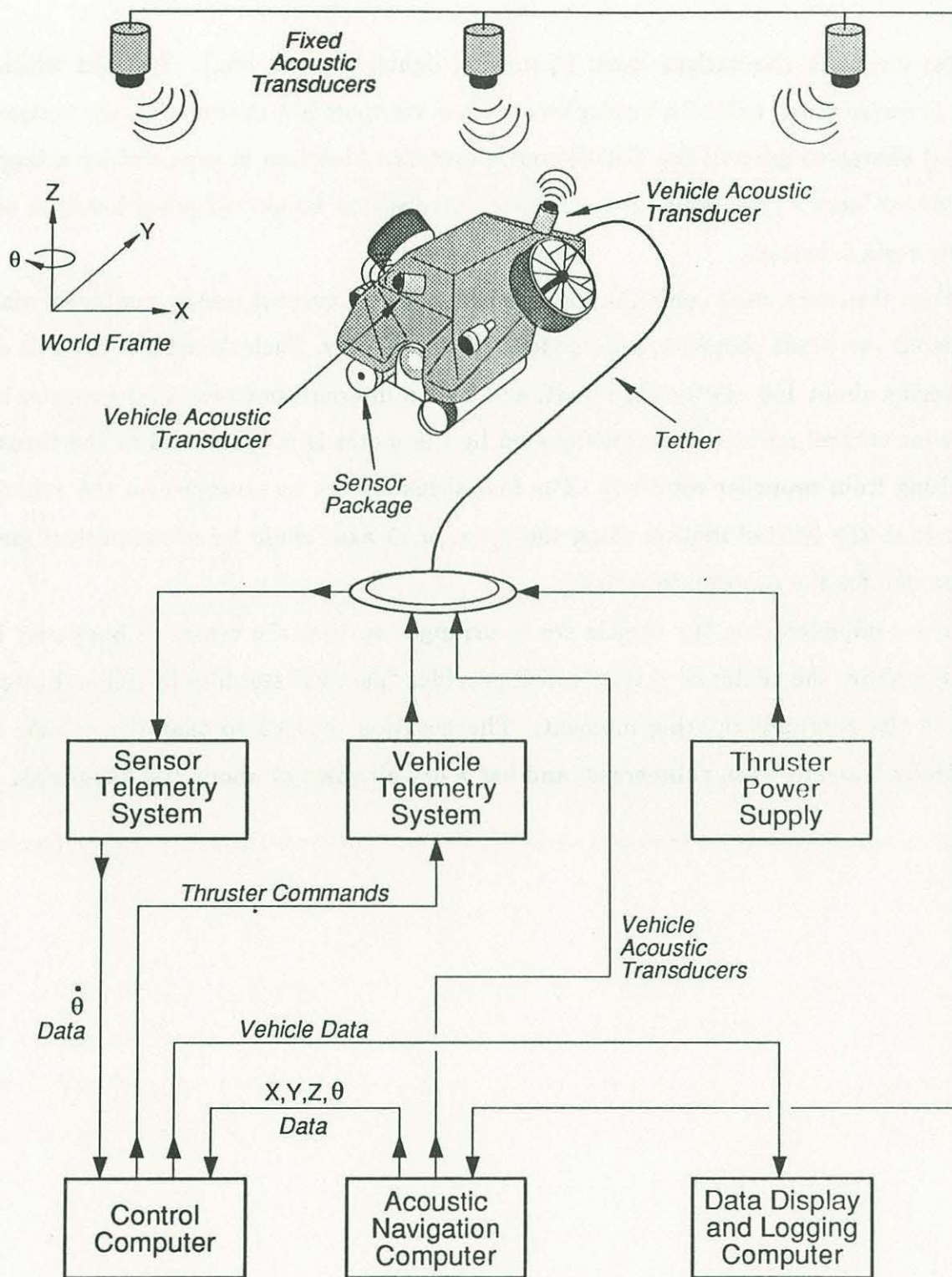


Figure 6.1 Experimental apparatus

6.1.2 Sensor Systems

Two sensor systems provide information about various vehicle parameters to the control system - an acoustic tracking system and a vehicle sensor package. The acoustic tracking system uses a relatively high frequency (300 KHz) acoustic pulse which is transmitted alternately from the two transponders mounted on the vehicle. Three similar transponders (acting as receivers) detect the transmitted pulse and allow for the calculation of transmission time (and hence range). An algorithm is then used to compute the vehicle position relative to a fixed (world) coordinate system. Two transponders were used on the vehicle to provide a vehicle heading reference with respect to the fixed coordinate frame - this was done by computing the angle between successive fixes, where the fixes alternated between the two vehicle transducers. The high frequency of the acoustic signal used minimizes the "multipathing" (different transmission paths resulting in an ambiguous travel time) problem and permits accuracy on the order of several centimeters. This system operates at update rates as fast as 10 Hz with a maximum range of about 100 meters.

The vehicle is also fitted with a sensor package which contains x , y , and z (body referenced) accelerometers, pitch and roll inclinometers (providing angular measurements) and rate sensors, yaw (rotation about the body z axis) rate gyroscope, and magnetic compass. The magnetic compass was found to be susceptible to interference in the metallic test tank used for the experiments and prompted the use of the dual transponder method to determine vehicle heading. The pitch and roll inclinometers were used to correct the acoustic navigation fixes (obtained with reference to the transponders on the top of the vehicle) for vehicle pitch and roll movements.

6.1.3 Telemetry and Tether System

The telemetry and tether system allows for communication and power transmission

between the control system hardware and the vehicle and its sensor systems. Commands to the thrusters are encoded onto a coaxial cable within the tether that is shared with the video signal from the camera. A small processor running at 30 Hz within the vehicle pressure vessel samples the digital commands and provides conversion to analog control voltages for the thrusters. The 120 volt D.C supply for the thrusters is also carried within the tether.

The sensor package has an internal processor and analog-to-digital converter which samples the individual sensors at 10 Hz. The sensor data is sent to the control computer via a dedicated 9600 baud serial communication line.

Each acoustic transponder communicates along an individual coaxial link to the acoustic navigation computer. This computer synchronizes the transmitter and receiver of the transponders so that travel times can be computed.

6.1.4 Control System Hardware

As shown in Figure 6.1, three separate computer system (6 MHz, 80286 processor-class) were used for control, acoustic navigation, and data display and logging. The control computer receives sensor package data from the vehicle and processed acoustic navigation data from the acoustic navigation computer. This data is used in the control algorithm to generate thruster commands which are sent to the vehicle via the telemetry system. The control computer also sends system parameters via a serial communications link to the data display and logging computer.

The acoustic navigation computer causes an acoustic pulse to be transmitted from (alternately) one of the vehicle transponders. This pulse rate is synchronized with the repetition rate of the control algorithm such that a new fix is generated for each cycle of the algorithm. This computer detects the received pulse at each of the fixed transponders and computes a travel time which is in turn converted to range using the speed of the pulse in water. An algorithm is then used to compute the applicable

(vehicle) transponder position relative to a fixed (world) coordinate frame. As mentioned earlier, successive fixes are obtained alternately from each vehicle transponder so that an absolute (world) heading can be obtained. The navigation data, consisting of (world) x , y , z , and Θ (heading) are sent via a serial communications link to the control computer for further processing.

The data display and logging computer receives a serial data stream from the control computer and allows for the real-time display of system parameters such as state variables, control commands, model parameters, etc. A provision is also made such that the data can be recorded for later review and analysis.

6.2 Control System Software

The block diagram for the control system software is shown in Figure 6.2. After performing an initialization sequence, the main program loop is entered. The most recent sensor data from the acoustic navigation system and the sensor package are obtained by using a ring buffer (last-in-first-out) strategy. A simulation capability is also provided (which uses the vehicle model to replace the sensor data) for testing algorithms before implementation on the vehicle itself.

The state observation and filtering process is complicated by the fact that the acoustic navigation system provides world x , y , z , and Θ position information only, whereas the “body” (vehicle) referenced position and velocity is actually needed for the control algorithm. In order to use a state estimation scheme of the standard form, it is necessary to convert the world positions into body positions. To accomplish this, a “pseudo-body” coordinate frame was established. The plan view of this coordinate frame is shown in Figure 6.3, where the “SNAME” (Society for Naval Architects and Marine Engineers) convention (see [87, 88]) has been used in the pseudo-body frame (y positive to starboard, z positive downward). Note that the rotation angle in the world and pseudo-body frame are identical. That is, the pseudo-body frame rotates along with the vehicle although its origin remains coincident with that of the world frame.

The coordinate transformation can be succinctly expressed in matrix form:

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} \cos(\Theta_B) & -\sin(\Theta_B) & 0 \\ -\sin(\Theta_B) & -\cos(\Theta_B) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} \quad (6.2.1)$$

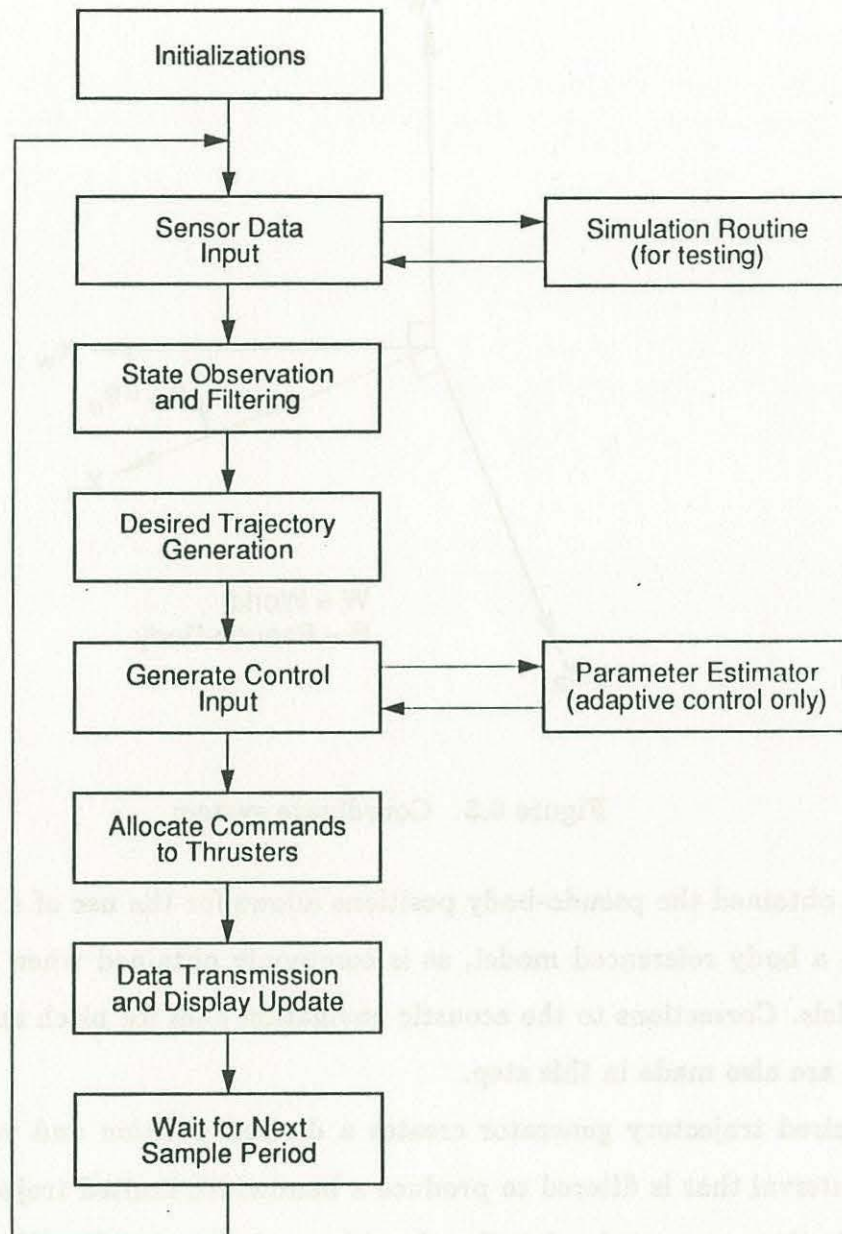


Figure 6.2 Control system software block diagram

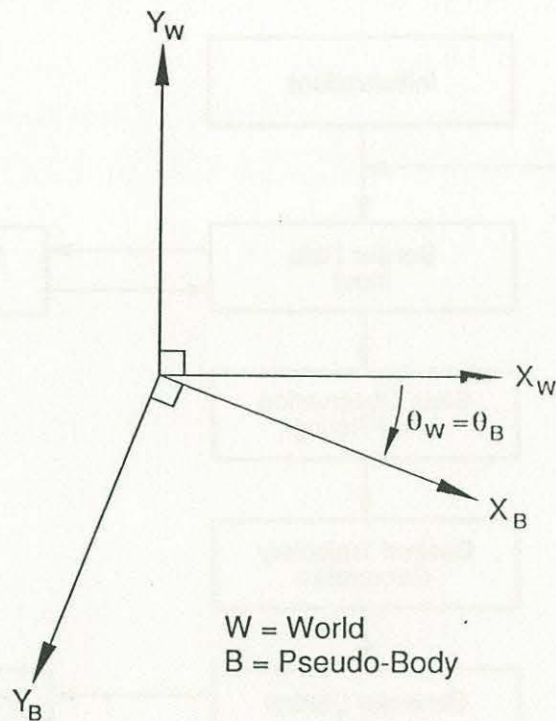


Figure 6.3 Coordinate system

Having obtained the pseudo-body positions allows for the use of a state estimator which uses a body referenced model, as is commonly obtained when using hydrodynamic models. Corrections to the acoustic navigation fixes for pitch and roll effects of the vehicle are also made in this step.

The desired trajectory generator creates a desired position and velocity for each sampling interval that is filtered to produce a bandwidth limited trajectory. The control input is then generated using the algorithms of Chapters 3 and 4. A parameter estimation routine is used for the adaptive control algorithms. Immediately after computing the required body referenced control inputs, the required thruster commands to achieve these inputs are calculated and issued through the telemetry system. Data transmission to the display and data logging computer follows, and the program enters a wait state until the next sample period begins.

Throughout the control system software it is also necessary to account for the $0 \rightarrow 360^\circ$ (or vice versa) discontinuity in Θ which occurs when the vehicle heading is close to "north" (0°) in the coordinate system. This is especially important in the state observation and trajectory generation functions to ensure smooth operation near this discontinuity.

6.3 Vehicle Modeling

6.3.1 System Identification

As explained in Section 5.1, two approaches are available for determining a discrete time model for a physical system - system identification or discretization of a continuous time model. Because of the uncertainty concerning the continuous time model structure that describes JJ (due to the hydrodynamic effects), it was decided to use the empirically based system identification approach.

The first decision to make before performing system identification is model structure to be used. A major component of this structure is the model order - a second order system was selected because it represented a good tradeoff between the ability to describe the system accurately yet not be overly cumbersome to manipulate. The two states selected corresponded to that of the sensors (or observer states) available - position and velocity.

To gain insight into the details of the model structure and to determine the appropriate regressor variables, the steady state velocity resulting from various levels of thruster inputs throughout the expected operating range was measured. The velocity was selected (rather than position) as the dynamics of the vehicle principally depend on hydrodynamic effects which are a function of vehicle movement.

Figure 6.4 shows the steady state velocity in the yaw degree of freedom (rotation about the vertical axis) resulting from a steady state thruster input in digital "counts" (± 2048 counts is the full range of the thruster). There is clearly a nonlinear relationship between the thruster input and resulting velocity, although it is not obvious whether the nonlinearity is due to increasing drag at higher velocities or due to decreasing thruster effectiveness at higher velocities. By experimenting with various nonlinear structures, it was found that the following form worked adequately:

$$x(k+1) = fx(k) + \frac{bu(k)}{1 + c |x(k)|} \quad (6.3.1.1)$$

where $x(k)$ is the velocity and f , b , and c are parameters to be determined. To use standard parameter estimation techniques it is necessary to have a model linear in the parameter set. The structure of (6.3.1.1) can be made linear in the unknown parameters by determining a value for c experimentally (this was found to be 0.05 for the yaw degree of freedom, for example).

A system identification experiment was done (again in the yaw of freedom) in which an input command consisting of three linearly combined sinusoids (of frequency 0.15, 0.3, and 0.5 radians/sec) was applied to the vehicle. These particular frequencies were used as they constituted the probable frequency range of desired trajectories in which the vehicle would be operating. Parameters were obtained using a batch least squares algorithm using a nonlinear structure (corresponding to (6.3.1.1) with $c=0.05$) and for a linear structure:

$$x(k+1) = fx(k) + bu(k) \quad (6.3.1.2)$$

where f and b are again parameters to be determined. One measure of the effectiveness of the resulting model obtained is to apply it to a second data set in simulation. Figure 6.5 shows the result of applying both nonlinear and linear models to the steady state data presented in Figure 6.4. The nonlinear model is seen to predict the actual vehicle behavior more accurately than the linear model over a wider velocity range. It is interesting to consider the implications of using the linear instead of nonlinear model for control design. From Figure 6.5, the linear model is seen to underestimate the thruster effectiveness at low velocities and overestimate the thruster effectiveness at high velocities. Use of the linear model at low velocities will thus result in a higher

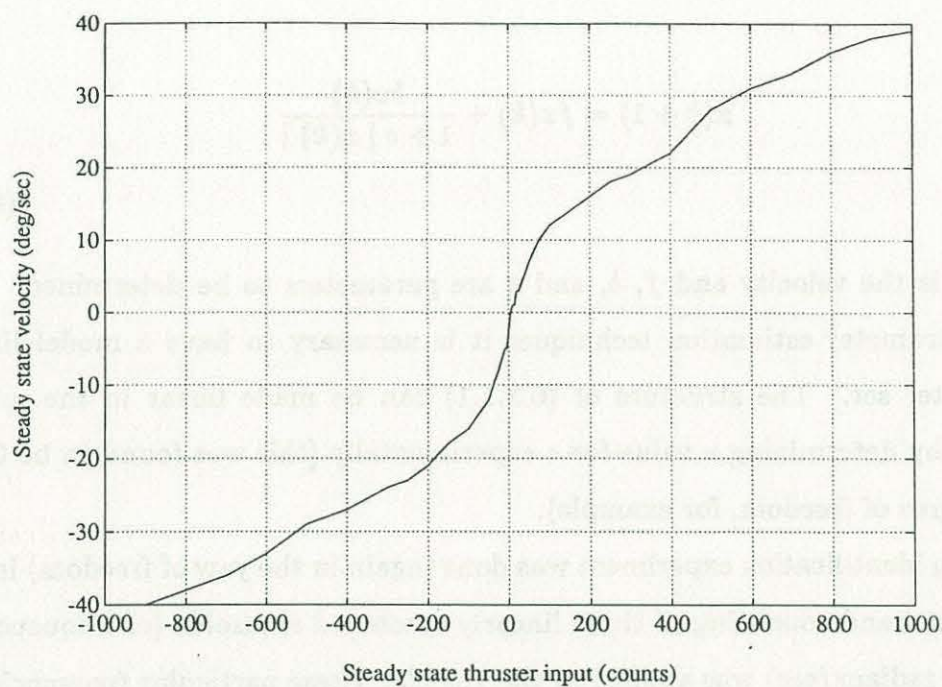


Figure 6.4 Steady state performance in the yaw degree of freedom

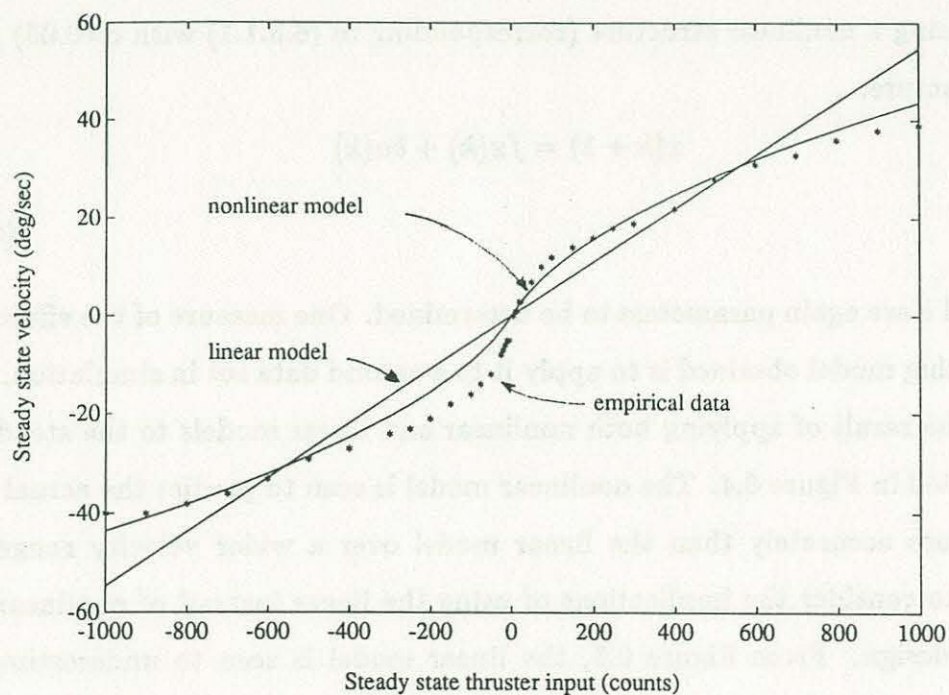


Figure 6.5 Model validation in the yaw degree of freedom

actual bandwidth being achieved than specified in the design, and the converse at higher velocities. The control bandwidth is normally limited by some external factors (see Section 5.4.) which implies that the "design" bandwidth must be made lower than this limit such that the actual bandwidth which results is at or below the limit. For example (again using the data of Figure 6.5), by making the design bandwidth lower than the limit, the actual bandwidth can be adjusted to be equal to the limit. However, at higher velocities (where the thruster effectiveness has been overestimated) the use of the lower design bandwidth results in a much lower actual bandwidth being achieved than is possible. The advantage of using the nonlinear model is that, by explicitly accounting for the nonlinear behavior, the design bandwidth can be set closer to the (theoretical) limiting bandwidth and result in the actual bandwidth more nearly remaining close to this limit throughout the entire velocity range.

Models for each of the four other "actively" controlled degrees of freedom (x, y, and z) were similarly obtained. To maintain simplicity in the resulting controllers, the models were uncoupled between degrees of freedom.

Concentrating on the nonlinear model, a general disturbance term was added the structure of (6.3.1.1) to account for sensor noise, thruster dynamics, disturbances, etc. The system structure therefore takes the form:

$$x(u+1) = fx(k) + \frac{bu(k)}{1+c|x(k)|}u(k) + d(k) \quad (6.3.1.3)$$

where parameter c is specified and $d(k)$ is a general disturbance term.

6.3.2 Model Imperfections

The model structure obtained in the previous section has admittedly been simplified a great deal to facilitate its use in the control system design. To apply the control methods of Chapters 3 and 4, it is necessary to consider the imperfections that exist in the model used for design purposes.

The largest modeling error results from the use of the thrusters as an input source. Each thruster has a "deadband" region for commands centered around zero where no thrust results. This deadband behavior arises from "stiction" associated with the motor shaft seals and from the torque control electronics located within the motor. The commanded input was augmented to eliminate most of this effect, but an ambiguity remained of about 2% of full thruster output. Although this appears to be a rather small magnitude, the vehicle was somewhat overpowered which meant that only about 20% of the thruster capability was normally used. The internal control electronics and hydrodynamic effects associated with the propeller caused additional dynamics between the commanded input and actual thrust delivered - these dynamics were estimated to be of a first order linear nature with a cut off frequency of about 6 radians/sec.

The tether force acting on the vehicle is a non-deterministic effect which influences the model. This is because the tether takes on random orientations depending upon the present as well as the past movement history of the vehicle. Time-variant hydrodynamic effects like vortex shedding, "added" mass, and propeller efficiency further compromise the model. Sensor noise must be accounted for as well as time delays that result from obtaining sensor information through the telemetry system. The telemetry system time delay also affects when control commands issued by the control computer actually reach the thrusters.

As mentioned earlier, the vehicle is passively stabilized in roll and pitch using a center of buoyancy - center of gravity righting moment. This results in a moderately damped oscillatory mode in both pitch and roll which changes the line of action of the thrusters and causes the top mounted acoustic transponders to oscillate from side to side.

The method used for dealing with these issues during the experimental tests and their implication on vehicle performance will be covered in the next section.

6.4 Experimental Results

The experimental results reported here will be confined to those performed in the yaw degree of freedom. These can be considered as indicative of the behavior of the other degrees of freedom as well, as the vehicle response is similar in each.

As detailed in Section 6.3.1, both linear and nonlinear models were formulated for the velocity behavior of the vehicle. For these experiments, an integrated velocity signal was used for the "position" or lower order state. The two dimensional model structure was therefore of the form:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & f \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{b}{1+c|x_2(k)|} \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d(k) \quad (6.4.1)$$

where $x_1(k)$ is the integrated velocity, $x_2(k)$ the velocity, T the sampling interval, $d(k)$ the disturbance term, and f, b, c parameters.

The sampling in this application was limited by the "reverberations" present in the (steel) experimental test tank from the acoustic navigation system. That is, a certain waiting period was necessary between transmitted pulses to allow for the acoustic energy to dissipate so as not to cause a false return at the receiving transponder. This limited the sampling interval to 0.22 seconds for a sampling rate of about 4.5 Hz. This was coincidentally very close to the control computer's maximum capability when executing the control system software of Section 6.2 simultaneously for four degrees of freedom. Thus the T of (6.4.1) was set to 0.22.

The system identification procedure described the "best fit" (in a least squares prediction sense) parameter values: $f = 0.95$, $b = 0.0066$, and $c = 0.05$. These will be recognized as corresponding to values used in the examples provided in earlier chapters.

A bound was determined for the disturbance term $d(k)$ based on the model imperfections discussed in Section 6.3.2. To reduce the relative effect of the thruster

deadband (which is a fixed percentage of total available motor torque), the thrusters were intentionally "degraded" by restricting the flow through the propellers so that the motors would be operating at a higher torque value. This results in a disturbance approximately bounded by 0.15 in the units of (6.4.1). This value was augmented by 0.05 to account for the remainder of the imperfections discussed in Section 6.3.2, making the final bound on the disturbance term equal to 0.2.

The bandwidth of the controller can be selected using the guidance of Section 5.4. The most stringent bandwidth limitation arises from the telemetry system time delay that affects both sensor inputs and command control inputs. Since this delay can be as long as two sampling intervals, from (5.4.2) thus equates to a continuous time bandwidth limitation of approximately 0.75 radians/sec. to account for cumulative effects of actuator dynamics and higher order unmodeled hydrodynamic dynamics, the bandwidth was further lowered to 0.5 radians/sec. This bandwidth can be mapped into the discrete time domain using (5.4.4), and gives a value of $p = 0.89$ for the eigenvalue that should characterize the s dynamics.

As discussed in Section 2.6 concerning input time delays, the proposed method of dealing with these delays is difficult to apply to nonlinear systems because of the (in general) inability to formulate adequate "predictor" models. Further, a predictive type scheme loses much of its attractiveness when used in a high disturbance environment. For this reason it was decided that the input time delay would not be explicitly dealt with but rather accounted for by a reduction in bandwidth as described above. Similar predictive schemes can be attempted to rectify the sensor time delay problem, but are subject to the same difficulties in this application as the methods for dealing with input time delays.

6.4.1 Non-Adaptive Control Results

The methods of Chapter 3 can be applied to the system model as described in the previous section:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.11 \\ 0 & 0.95 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{0.0066}{1+0.05|x_2(k)|} \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d(k) \quad (6.4.1.1)$$

where $d(k)$ is bounded by 0.2. a small amount of uncertainty was added to f and b (1% and 15% , respectively) to account for possible errors in the system identification procedure or time-variant hydrodynamic effects. The controller is implemented exactly as in Example 3.3 using the appropriate values of $F(= 0.009 | x_2(k) |)$ and $b (=1.1.5)$ given the above uncertainties.

Using the value for p of 0.89 (corresponding to a continuous time bandwidth of 0.5 radians/sec.), a desired trajectory was specified for the velocity state $x_2(k)$ consisting of a linear combination of three sinusoids with frequencies 0.1, 0.2, and 0.3 radians/sec. The desired trajectory for the position state $x_1(k)$ was specified in accordance with the consistency requirement of Section 2.5.

Figure 6.6 shows the s and Φ trajectories obtained during the experimental run, which are in units of degrees (of rotation angle Θ) per second. The fact that s remains within the boundary layer Φ tends to support the modeling assumptions made concerning parameter values and imperfections present. The trajectory following performance for x_1 and x_2 are shown in Figures 6.7 and 6.8, respectively, and is seen to be quite satisfactory. The control input of Figure 6.9 is also shown to illustrate its bandwidth limited nature.

In an attempt to achieve even greater performance using a higher control bandwidth, it is reasonable to attempt to increase the 0.5 radians/sec. used in this experiment to, for example, 0.75 radians/sec. Figure 6.10 shows the s trajectory that results when using this higher bandwidth - some unmodeled dynamics are apparently being excited

(probably due to the telemetry time delays). The frequency of this behavior is seen to be about 3 radians/sec., which will become significant when using the adaptive controllers in the next section. Figure 6.11 illustrates the control input that results - the presence of s in the control law results in the high frequency, unmodeled dynamics affecting it as well. Thus, the original control bandwidth choice appears to have been a good one.

It is interesting to compare the performance obtained when using the nonlinear model to that when using the linear model (i.e. that of the structure (6.3.1.2) with additive disturbance term). The s trajectory resulting from this experiment is shown in Figure 6.12, and should be compared to the result using the nonlinear model as shown in Figure 6.6. The trajectory following performance using the linear model is similarly shown in Figures 6.13 and 6.14. Qualitatively, the nonlinear model-based controller appears to outperform the linear model-based controller. A more quantitative measure can be made by considering the standard deviation of the two s trajectories, which have been plotted together in Figure 6.15. The standard deviation of s can be computed for the entire experimental run for each model type, with the result being 2.58 for the linear model-based controller and 1.50 for the nonlinear model-based controller. By using the standard deviation as a performance measure, this implies that the nonlinear model-based controller offers about 72% better performance (on average for this desired trajectory) than that available when using the linear model-based controller.

The use of the non-adaptive controllers when the parametric uncertainty is much greater than considered here will be examined in the next section.

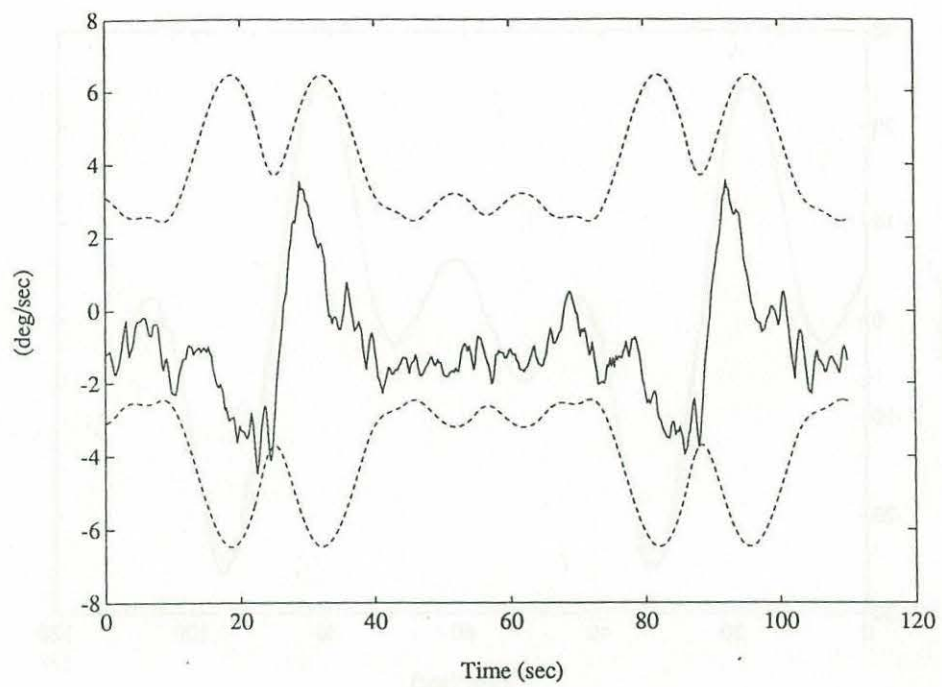


Figure 6.6 S (-) and Φ (- -)

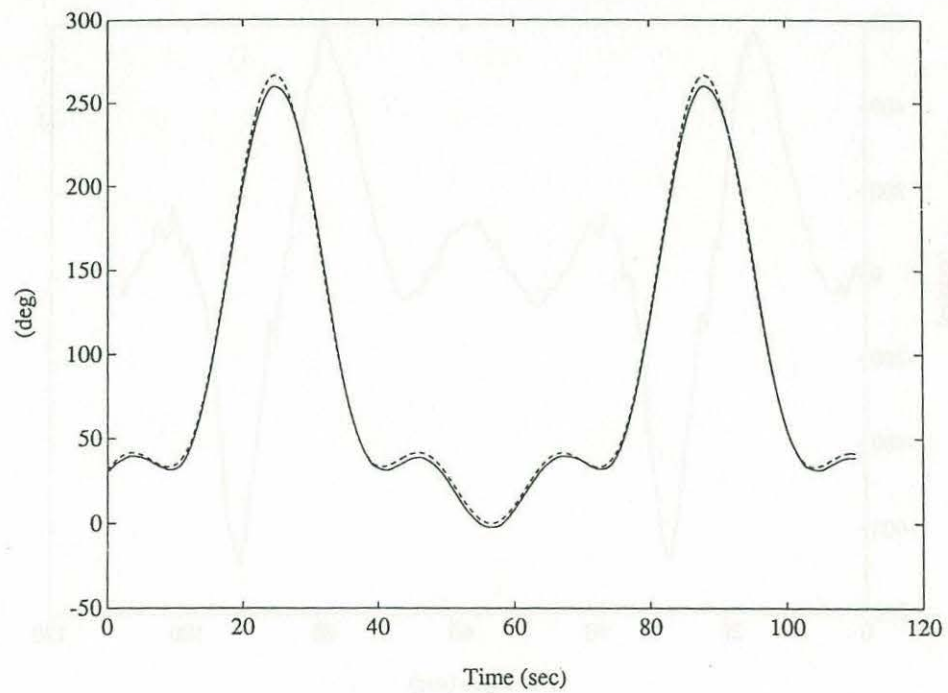


Figure 6.7 Actual (-) and desired (- -) x_1

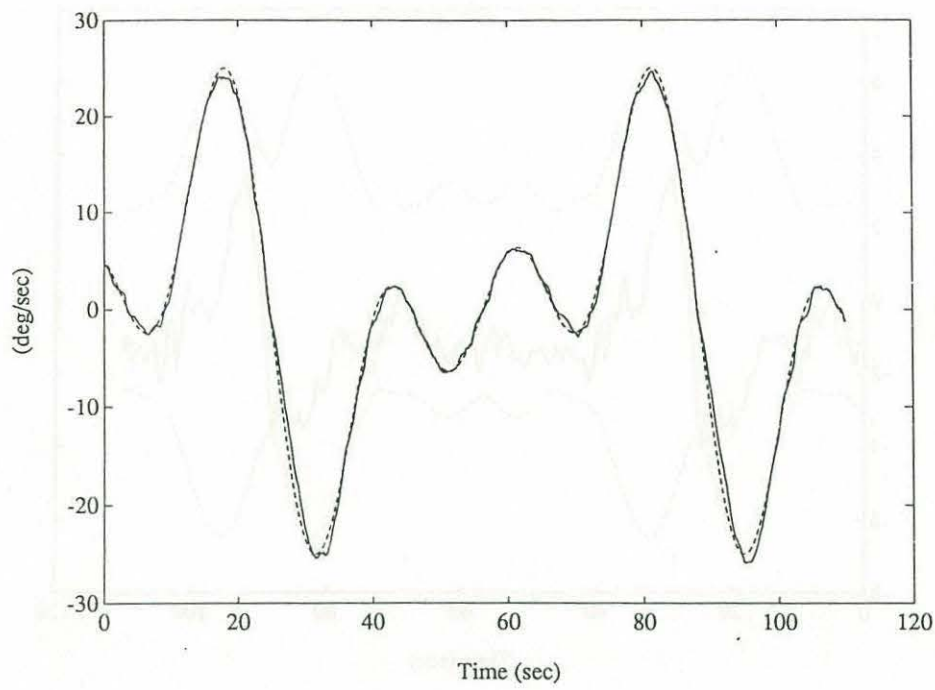


Figure 6.8 Actual (-) and desired (- -) x_2

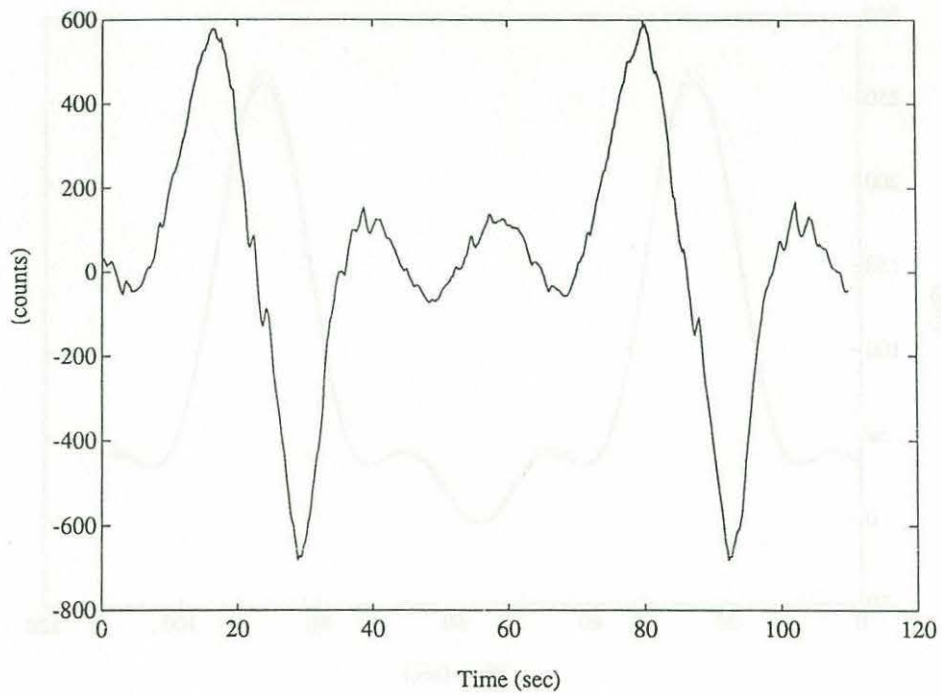


Figure 6.9 Control input

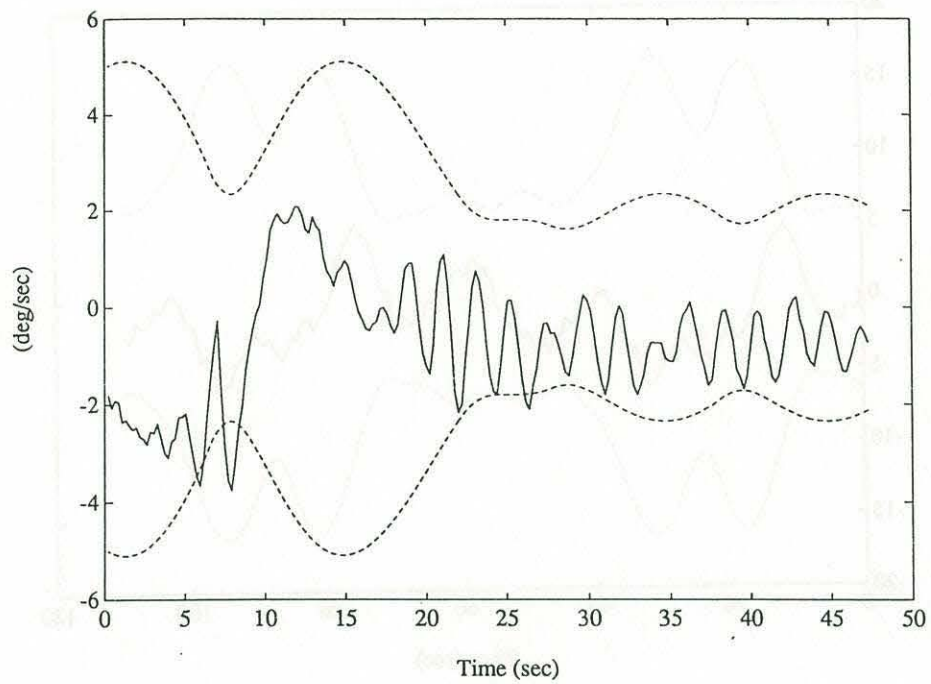


Figure 6.10 S (-) and Φ (- -)

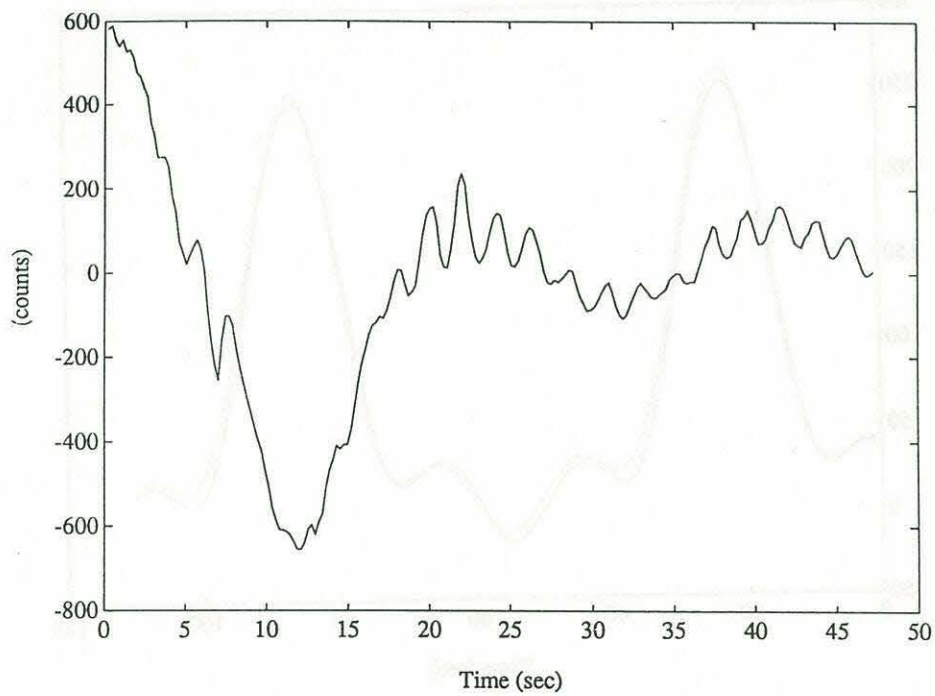


Figure 6.11 Control input

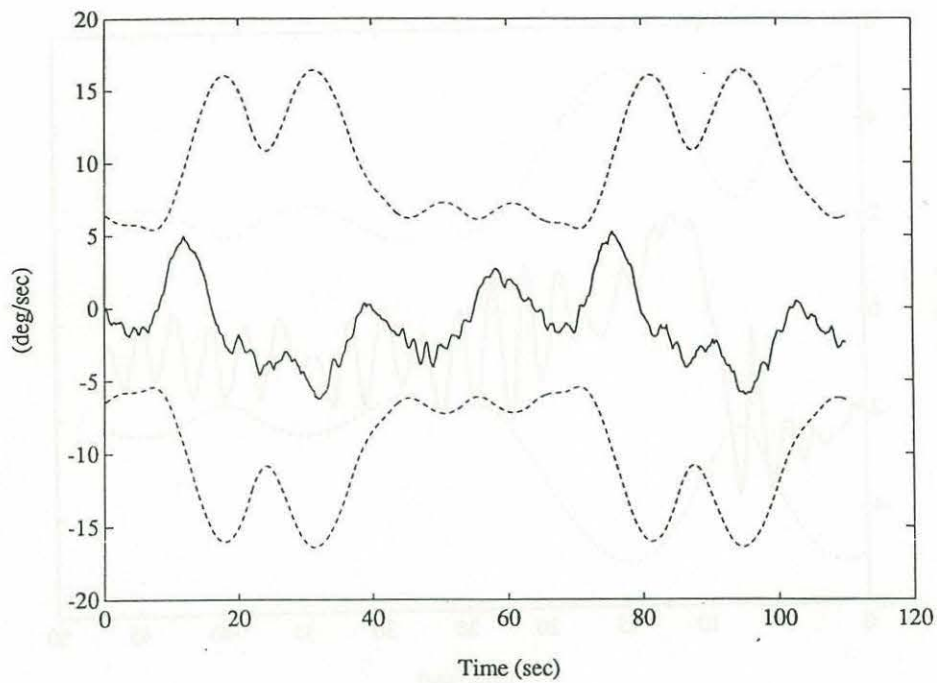


Figure 6.12 S (-) and Φ (- -) using a linear model

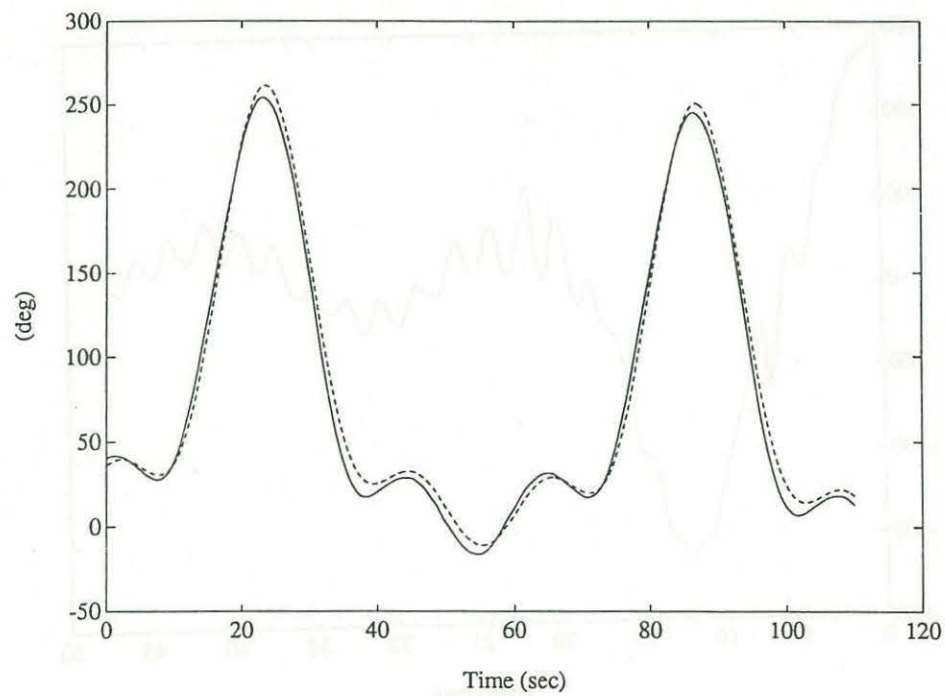


Figure 6.13 Actual (-) and desired (- -) x_1 using a linear model

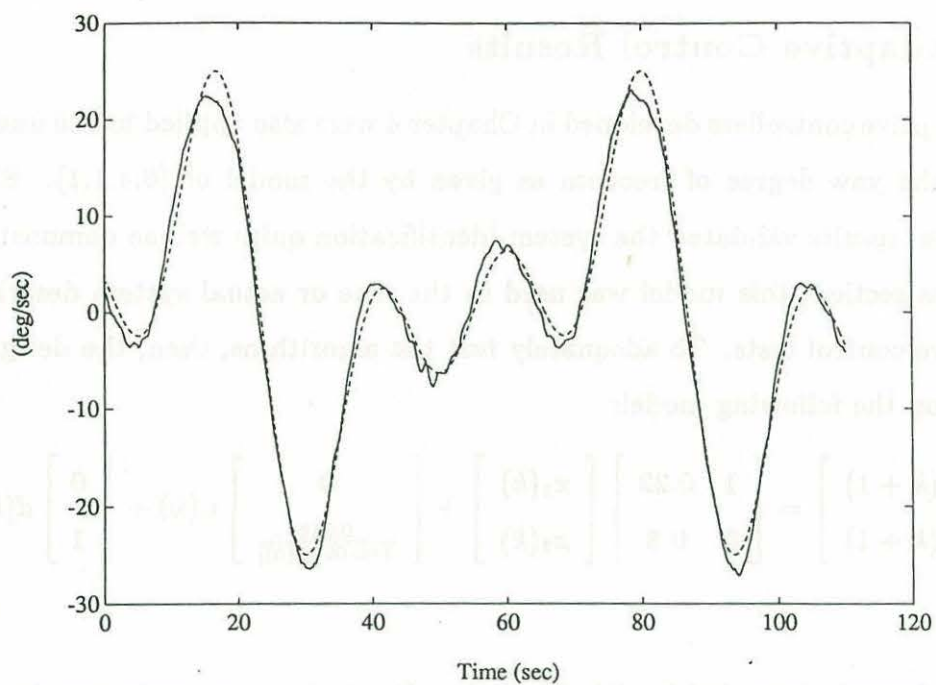


Figure 6.14 Actual (-) and desired (- -) x_2 using a linear model

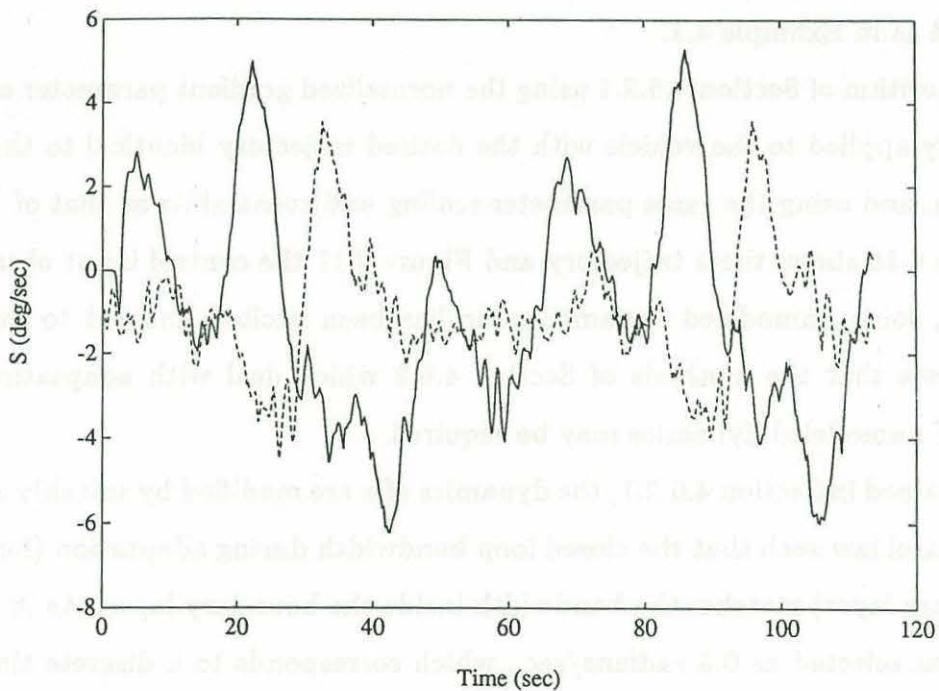


Figure 6.15 Controller performance using linear (-) and nonlinear (- -) models

6.4.2 Adaptive Control Results

The adaptive controllers developed in Chapter 4 were also applied to the underwater vehicle in the yaw degree of freedom as given by the model of (6.4.1.1). Since the experimental results validated the system identification quite well as demonstrated in the previous section, this model was used as the true or actual system description in the adaptive control tests. To adequately test the algorithms, then, the design model was based on the following model:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.22 \\ 0 & 0.8 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{0.012}{1+0.05|x_2(k)|} \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d(k) \quad (6.4.2.1)$$

where $d(k)$ is again bounded by 0.2. Note that (6.4.2.1) corresponds exactly with the model used in the Chapter 4 examples. For ease of reference, denote the value 0.8 as parameter 1 and the value $100 \cdot 0.012 = 1.2$ as parameter 2, where the parameters have been scaled as in Example 4.1.

The algorithm of Section 4.5.2.1 using the normalized gradient parameter estimator was initially applied to the vehicle with the desired trajectory identical to that of the last section, and using the same parameter scaling and constraints as that of Example 4.1. Figure 6.16 shows the s trajectory and Figure 6.17 the control input obtained.

Clearly, some unmodeled dynamic mode has been excited and led to instability. This suggests that the methods of Section 4.6.2 which deal with adaptation in the presence of unmodeled dynamics may be required.

As explained in Section 4.6.2.1, the dynamics of s are modified by suitably augmenting the control law such that the closed loop bandwidth during adaptation (i.e. outside the boundary layer) matches the bandwidth inside the boundary layer. As in Example 4.3, this was selected as 0.5 radians/sec., which corresponds to a discrete time eigenvalue of $p = 0.89$. To ensure that adaptation only occurs during relevant excursions outside the boundary layer, the methods of Section 4.6.2.2 are used to bound

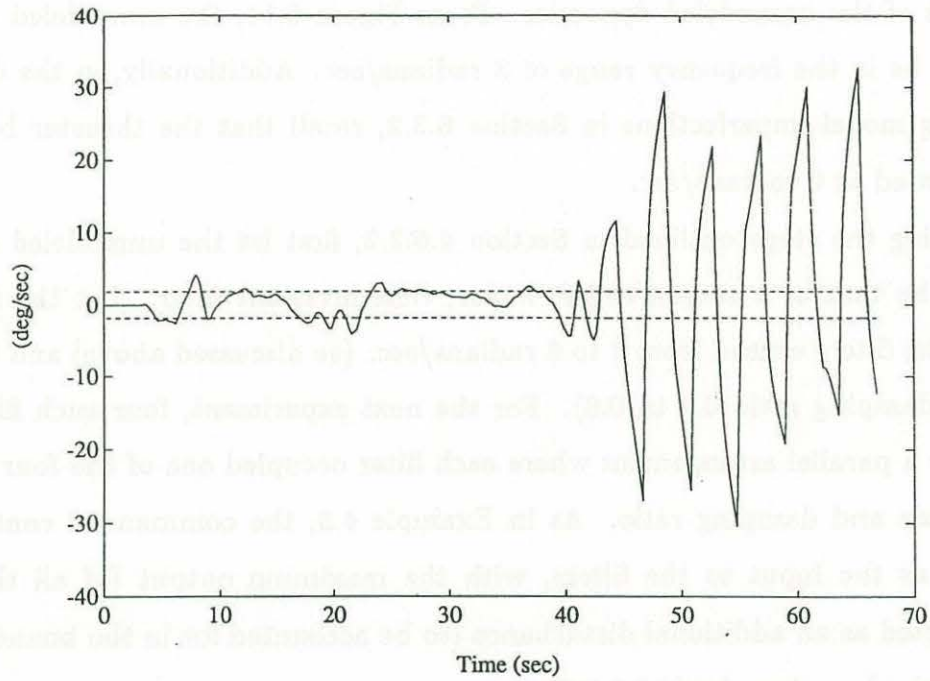


Figure 6.16 S (-) and Φ (- -)

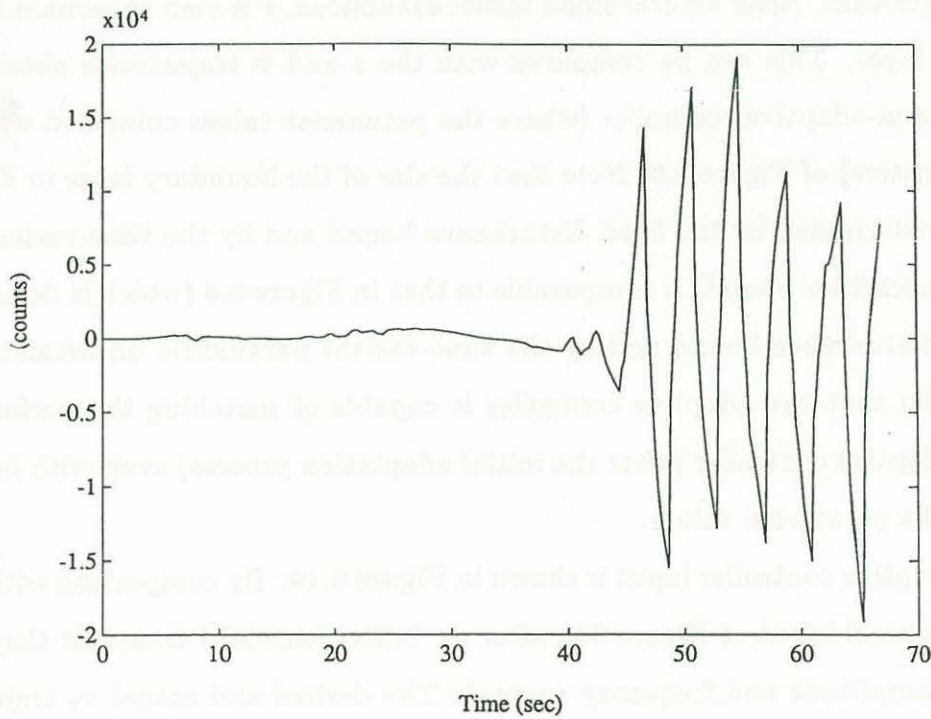


Figure 6.17 Control input

the effects of the unmodeled dynamics. From Figure 6.11, the unmodeled dynamics appear to be in the frequency range of 3 radians/sec. Additionally, in the discussion concerning model imperfections in Section 6.3.2, recall that the thruster bandwidth was estimated at 6 radians/sec.

Following the steps outlined in Section 4.6.2.2, first let the unmodeled dynamics structure be that of a second order, linear, time-invariant filter. Let the frequency range of the filters extend from 3 to 6 radians/sec. (as discussed above) and be lightly damped (damping ratio 0.3 to 0.6). For the next experiment, four such filters were utilized in a parallel arrangement where each filter occupied one of the four extremes in frequency and damping ratio. As in Example 4.3, the commanded control input was used as the input to the filters, with the maximum output (of all the filters) being selected as an additional disturbance (to be accounted for in the boundary layer determination) as given by (4.6.2.2.5).

Figure 6.19 illustrates the s and Φ trajectories obtained when using the modified control algorithm. After several small initial excursions, s is seen to remain within the boundary layer. This can be compared with the s and Φ trajectories obtained when using the non-adaptive controller (where the parameter values coincided with the actual parameters) of Figure 6.6. Note that the size of the boundary layer in Figure 6.18 (which is determined by the fixed disturbance bound and by the time-variant unmodeled dynamics filter bound) is comparable to that in Figure 6.6 (which is determined by the fixed disturbance bound and by the time-variant parametric uncertainty bound). This implies that the adaptive controller is capable of matching the performance of the non-adaptive controller (after the initial adaptation process) even with large initial errors in the parameter values.

The adaptive controller input is shown in Figure 6.19. By comparison with the non-adaptive control input of Figure 6.9, after an initial (smooth) transient they are very similar in amplitude and frequency content. The desired and actual x_2 trajectories in Figure 6.20 illustrate how trajectory following performance increases as the adaptation

process progresses.

The parameter estimator behavior is shown in Figures 6.21 and 6.22. Note that parameter changes only occur when s moves outside the boundary layer, and that parameter estimation is essentially complete after 70 seconds. The norm of the parameter error vector is plotted in Figure 6.23 to demonstrate that the decreasing parameter error norm property of the normalized gradient parameter estimation algorithm has been preserved despite the presence of disturbances and unmodeled dynamics. As shown in Example 4.1, the desired trajectory was sufficiently rich such that parameter convergence outside the boundary layer was assured. It is also important to notice that the parameters do not converge exactly to their (believed) actual values. This is due to the presence of disturbances and unmodeled dynamics (as defined by the boundary layer) as well as the probable time-variant behavior of the actual parameters over a small range (due to hydrodynamic effects).

The adaptive controller utilizing the least squares parameter estimation scheme of Section 4.5.2.2 with the special modifications for bandwidth control and relevant adaptation of Section 4.6.2 incorporated. Figures 6.24 and 6.25 show the parameter estimation behavior for the same initial parameter values used previously. This behavior is similar to that of the normalized gradient estimator of Figures 6.21 and 6.22, which implies that the (normally) faster convergence of the least squares algorithm is being offset by the controller modifications necessary to maintain stability.

It is interesting to compare the performance of the non-adaptive controller which is "robust" to a large parametric error with the adaptive controller. Figure 6.26 shows the s and Φ trajectories using the non-adaptive controller where parameter 1 is selected as 0.8 and is robust to a parametric error of ± 0.15 (thereby encompassing the actual value of 0.95) as in Example 3.2. Because of the large error in parameter 1, there is a large variation in s as predicted by the size of the boundary layer. The trajectory following performance can be observed in Figure 6.27, and is rather poor as expected. These are contrasted with the corresponding adaptive controller results shown in Figures 6.28

and 6.29. After initial adaptation, the s trajectory of the adaptive controller remains essentially within the boundary layer which is an order of magnitude less than that of the non-adaptive controller.

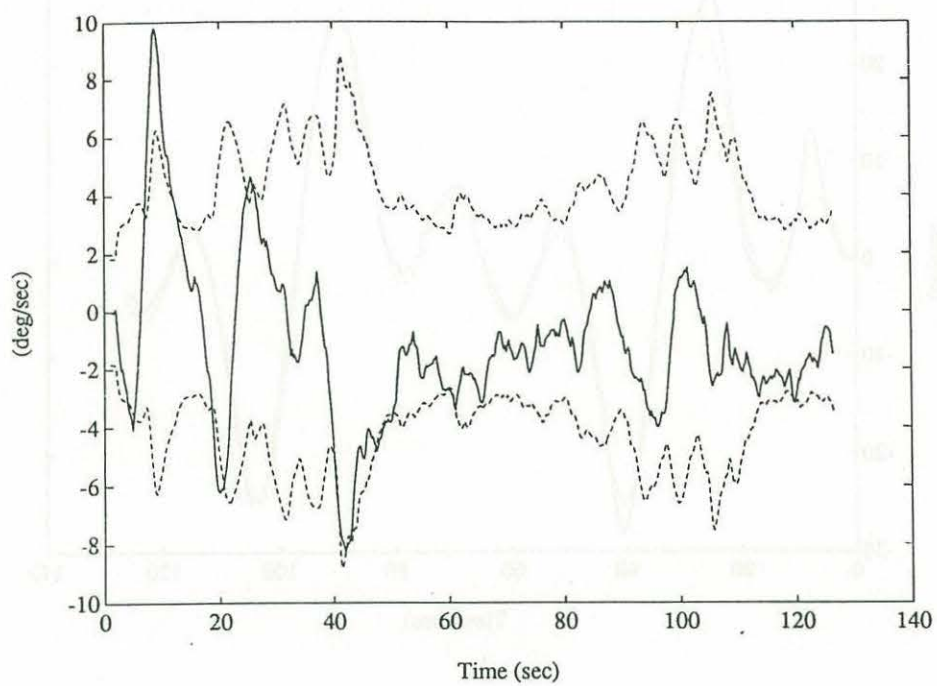


Figure 6.18 S (-) and Φ (- -)

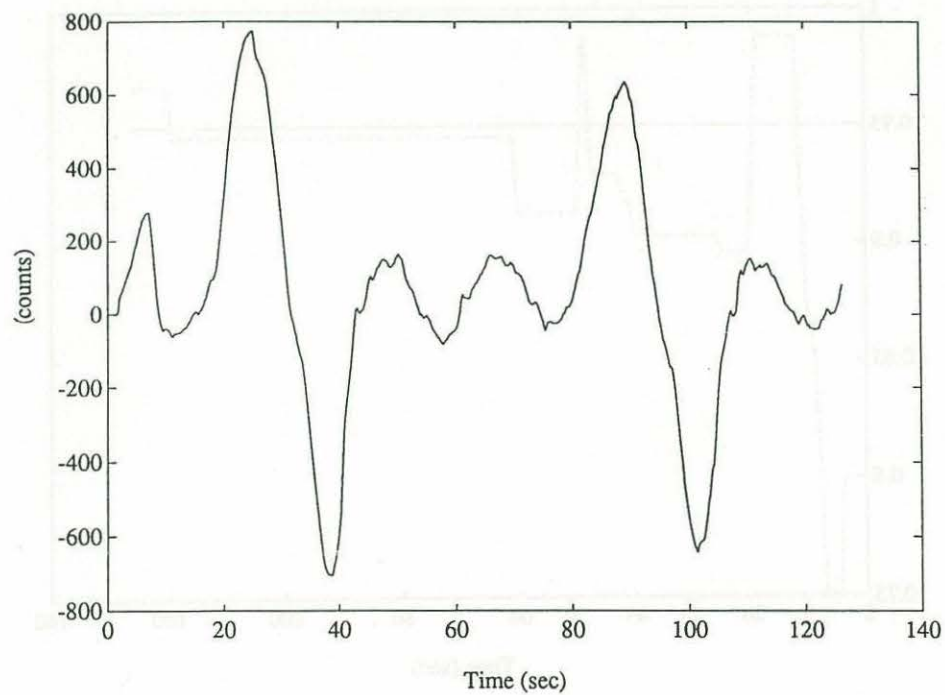


Figure 6.19 Control input

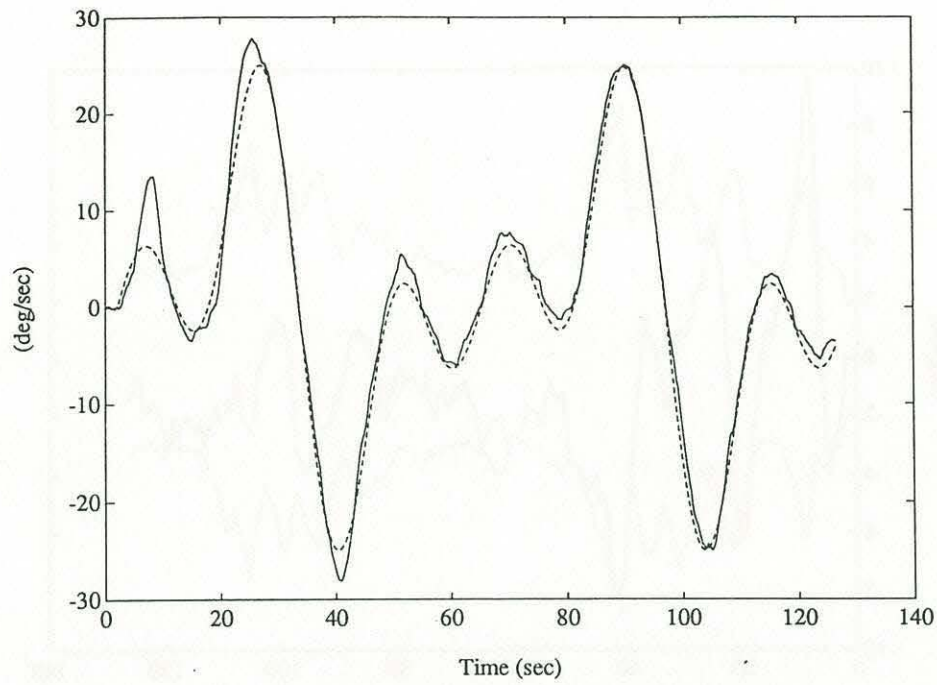


Figure 6.20 Actual (-) and desired (- -) x_2

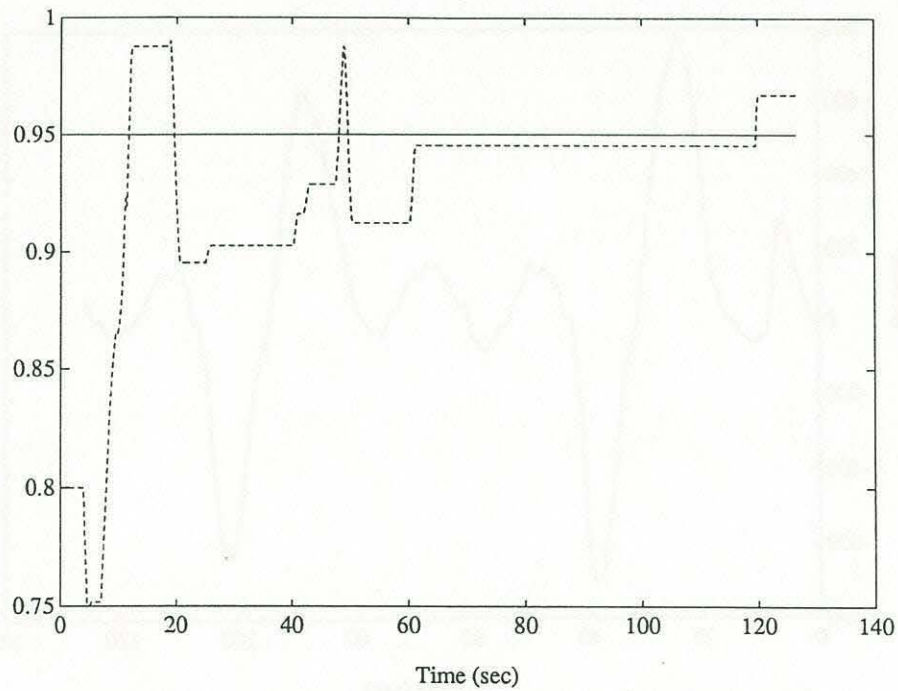


Figure 6.21 Actual (-) and estimated (- -) parameter 1

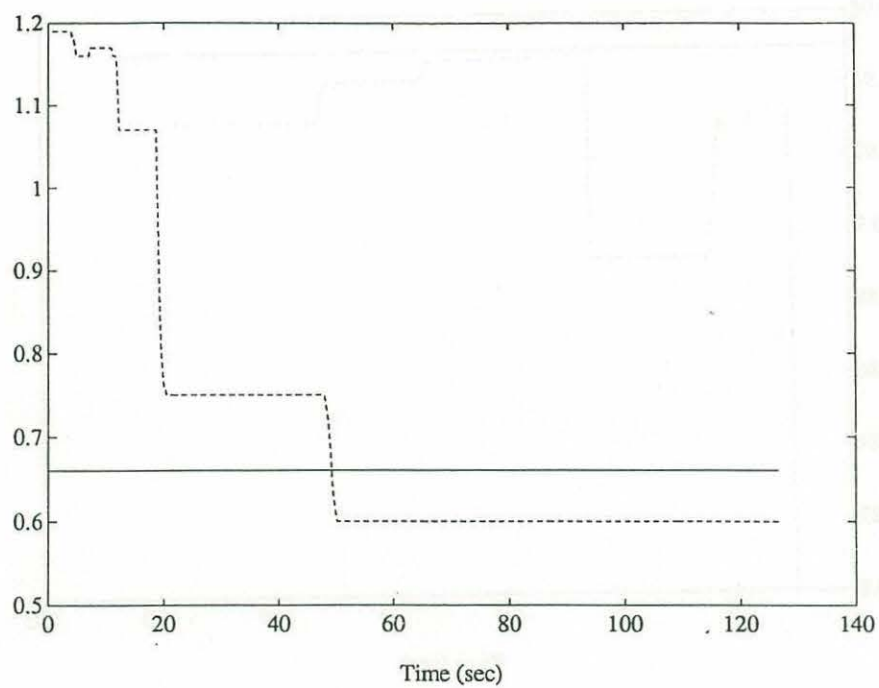


Figure 6.22 Actual (-) and estimated (- -) parameter 2

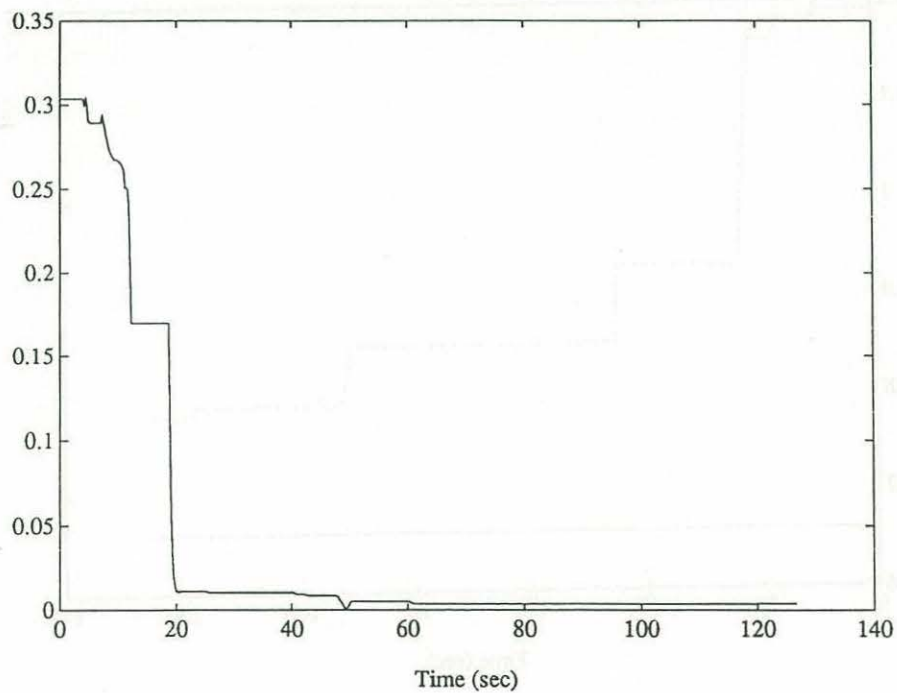


Figure 6.23 Parameter error norm

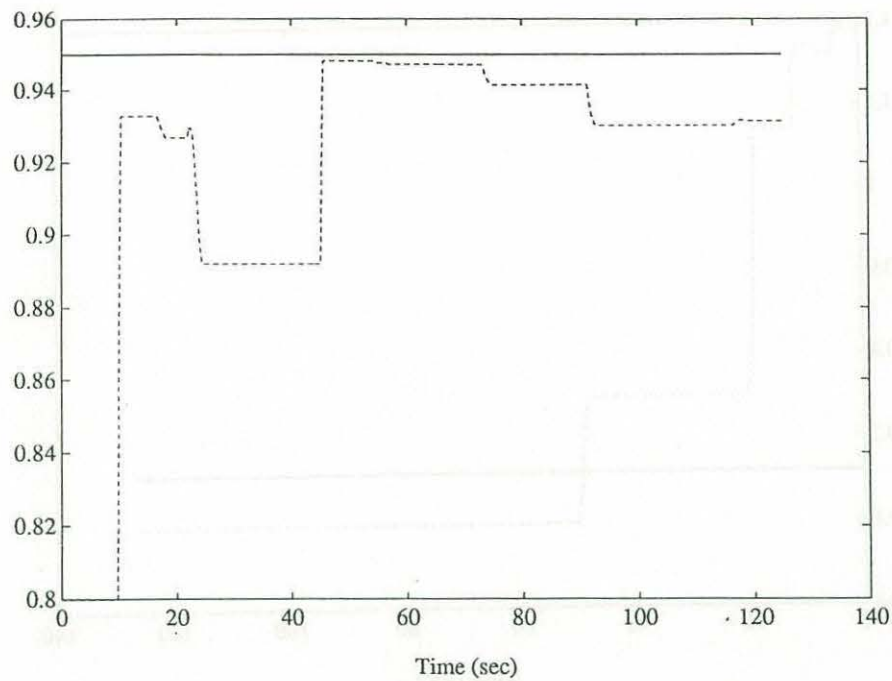


Figure 6.24 Actual (-) and estimated (- -) parameter 1

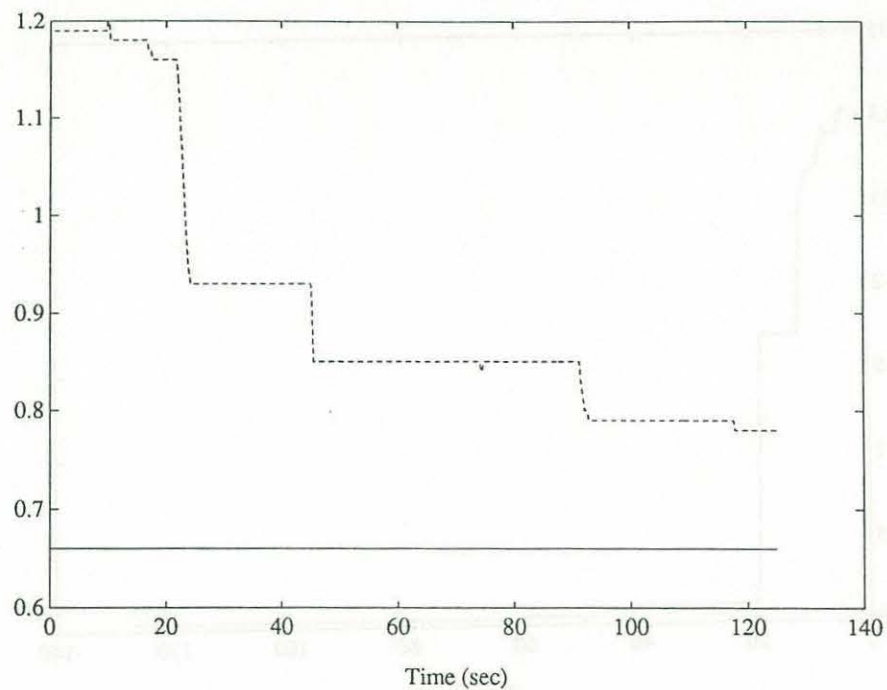


Figure 6.25 Actual (-) and estimated (- -) parameter 2

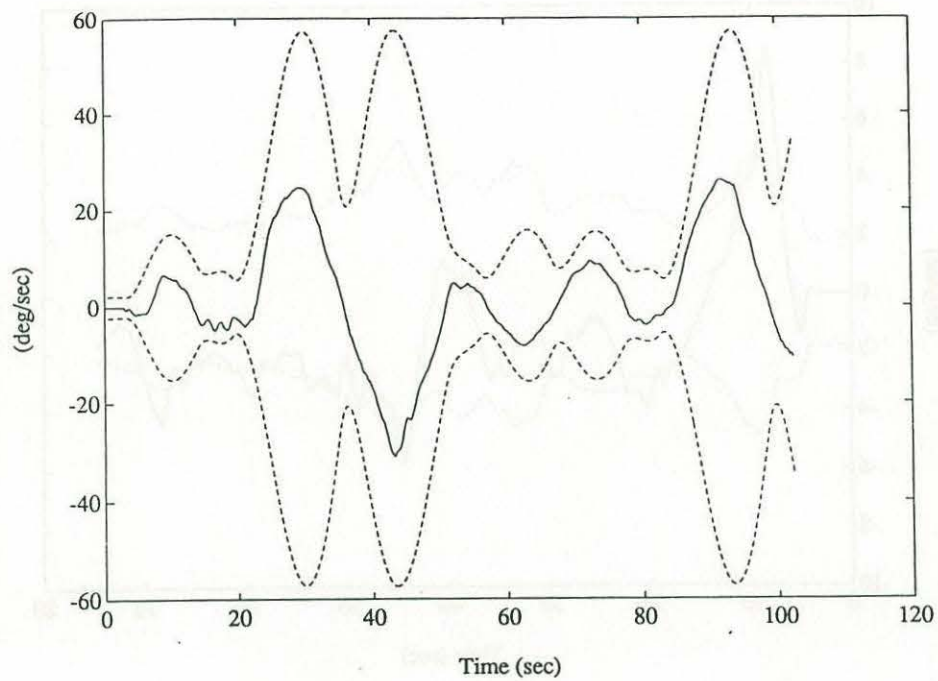


Figure 6.26 S (-) and Φ (- -)

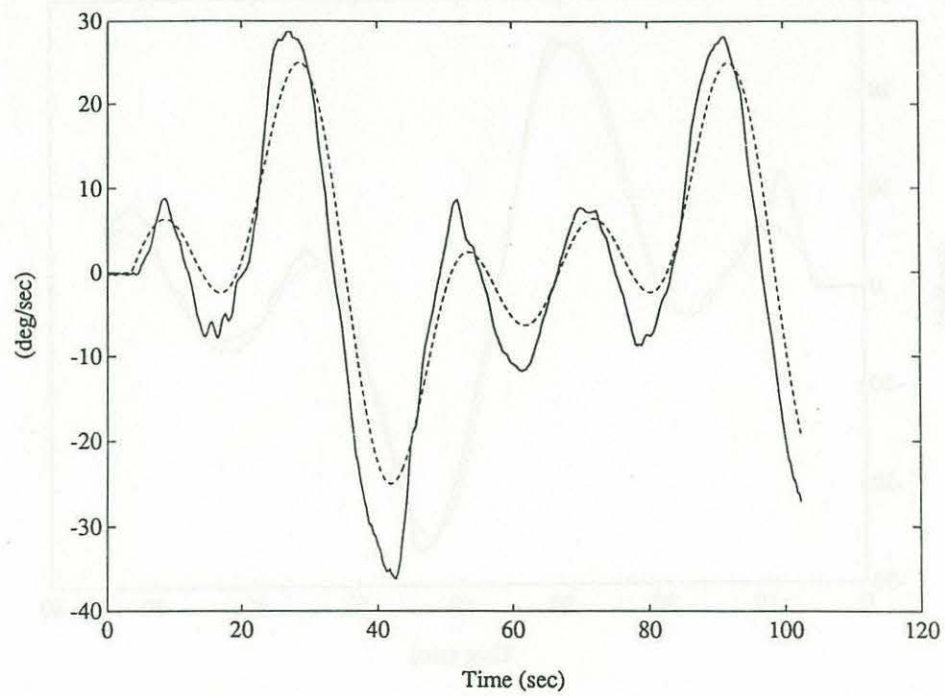


Figure 6.27 Actual (-) and desired (- -) x_2

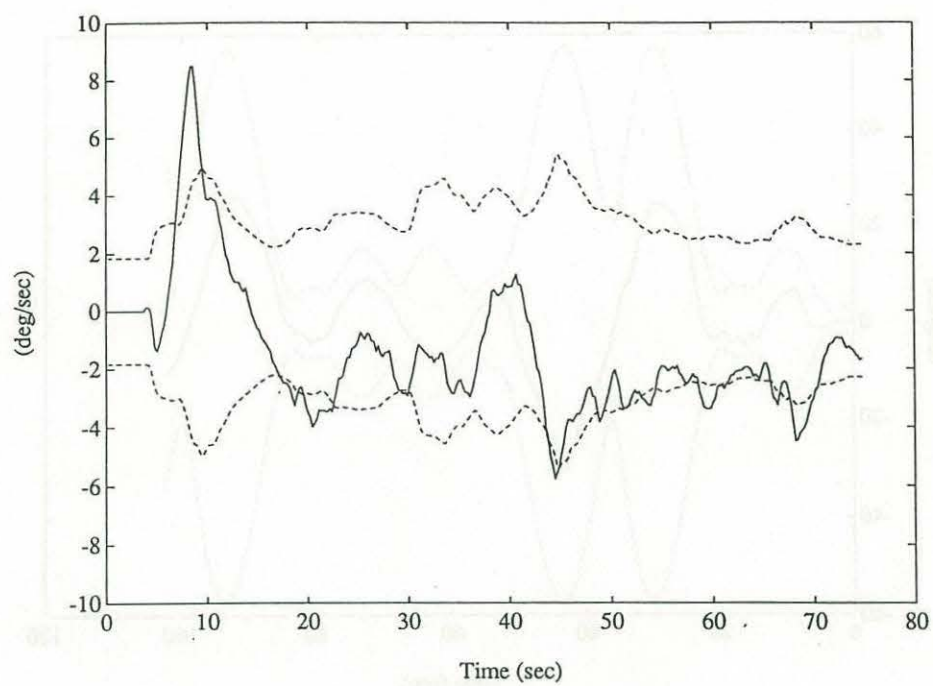


Figure 6.28 S (-) and Φ (- -)

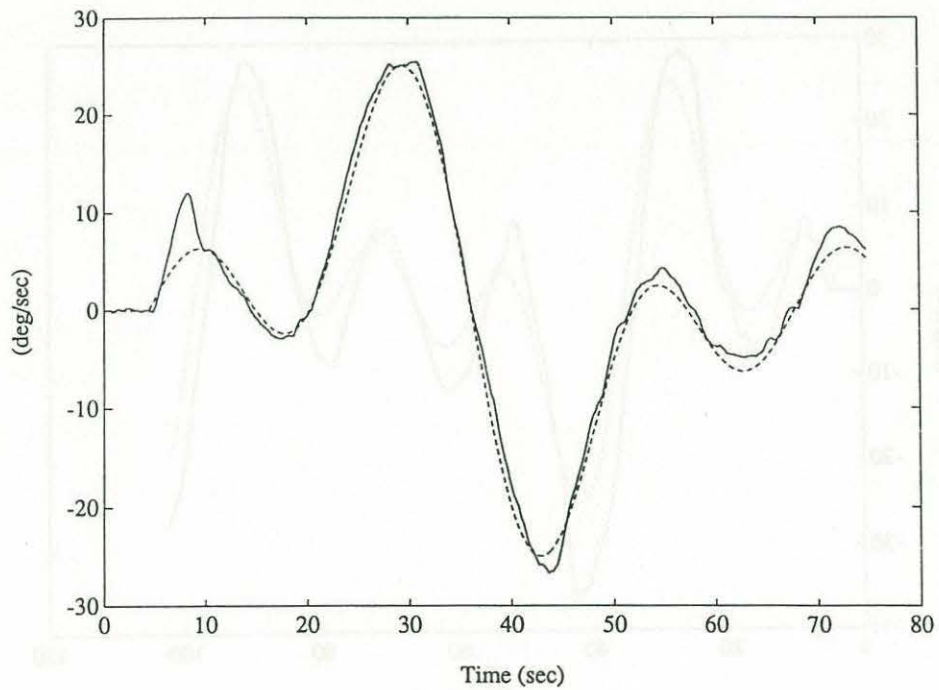


Figure 6.29 Actual (-) and desired (- -) x_2

Chapter 7

Summary, Conclusions, and Recommendations

Chapter 1 began with the motivation for this research work, namely to develop a control design methodology that would apply to the many challenges of remotely operated underwater vehicle control. These challenges, consisting of time-variant, nonlinear behavior and the lack of well known system models, are commonly encountered when dealing with physical system control in general and therefore can potentially be used in other applications as well.

Chapter 2 introduced the proposed methodology that was investigated in this thesis for dealing with these challenges - the use of a summarizing scalar along with discrete time sliding surfaces. This chapter was developed under the assumption of a perfectly known system without disturbances. The concept of a sliding surface using discrete time control was found to be similar to that of the sliding surface obtained using continuous time control, with the exception that in general it is only possible to remain within a bounded region of the surface rather than exactly on the surface. In discussing the selection of the sliding surface, a straightforward method was given for linear, time-invariant systems that allowed for the dynamics on the surface to be specified. The

methodology was also shown to apply to a restricted class of nonlinear and time-variant systems, where the nonlinear and time-variant behavior is confined to a single dimension (off the surface) which can be "cancelled" by the control input in order to maintain the system state within a bounded region of the surface. The controller formulation was found to extend naturally to the trajectory following form, which encompasses the trajectory tracking problem as well as the regulator problem. This formulation also resulted in a trajectory "consistency" requirement so that the specified (multi-dimensional) trajectory would be attainable at each sampling instant by the single dimensional control input. The discrete time formulation was found to be particularly well suited for handling input time delays that are integral multiples of the sampling period.

Chapter 3 introduced relaxations in the stringent assumptions from Chapter 2 by considering bounded disturbances, uncertainty in system dynamics and control gain, and acknowledging that the system model was a reduced order representation of the actual system. As in the case of nonlinear and time-variant behavior (even if perfectly known) from Chapter 2, the disturbances and other uncertainties were required to be constrained to a single dimension (again, off the surface). A major difference between the discrete and continuous time methodology is the nature of the stability proofs necessary when the relaxations dealing with uncertainty in system dynamics and control gain are included. This difference arises from the fact that the system operates in an open loop mode between sampling instants, and it is necessary to insure proper behavior during this time. The solution proposed involved a smoothing of the discontinuous control law in a boundary layer region surrounding the surface. A control law with linear feedback was used within the layer, and was able to show stability for a range of uncertainty in system dynamics and/or gain. The stability arguments for the relaxation dealing with disturbances were found to be similar to those used in the continuous time case, and were based on the boundedness of the disturbance inputs.

In Chapter 4 it was shown that if the system model could be parameterized in a

linear fashion, then the multi-dimensional control problem could be formulated into a single-dimensional parameter estimation problem. Parameter estimation techniques were reviewed and extended to the bounded disturbance case. These estimation schemes were then used in conjunction with the control algorithms of Chapter 3 to maintain stability in the face of parametric errors as well as disturbances. An advantage of the adaptive controller was its ability to deal with a (theoretically) unlimited range of parametric error unlike that of the non-adaptive controller which is restricted to a range of parametric error. The use of the adaptive controller, though, introduces additional states into the controller in a nonlinear and time-variant manner which makes the analysis of stability in the presence of unmodeled dynamics very difficult.

Two mechanisms were proposed as possible origins of instability in the closed loop adaptive system - that of bandwidth modification of the control law and the possibility of corrupted adaptation signals being used to drive the parameter estimator. These are presented in a rather heuristic manner because of the intractability of dealing with vaguely defined unmodeled dynamics coupled with a highly nonlinear system. Possible solutions are proposed to these mechanisms which involve modifying the control law to provide bandwidth control during the adaptation process and to ensure that only relevant information is used to drive the parameter estimation algorithm.

Chapter 5 gave guidance on the implementation of the methodology on physical systems. Considered first was the ability to formulate a discrete time model of a continuous time process. The field of system identification represents a well developed body of knowledge that provides a discrete time model given discrete data points that could result from an empirical test of a system. If a well structured continuous time system exists, then a method was shown for obtaining a suitable (but, in general, approximate) discrete time system model. Some differences between the use of the continuous time or the discrete time algorithm were discussed. It was decided that the insight provided by the continuous time formulation could be a powerful advantage, but slow sampling rates (relative to the closed loop bandwidth) could result in an

unpredictable and/or unstable system when implemented in a discrete time manner. The discrete time formulation, in contrast, is designed explicitly with the discrete implementation in mind but does not offer the physical insight of the continuous time formulation.

The system form was found to be a limitation of the sliding surface method as only permits the presence of nonlinear time-variant, uncertain behavior in a single dimension. Although a transformation approach could possibly convert a known general nonlinear system to, for example, controller canonical form, there appears to be no similar method for uncertain systems. This is because the resulting uncertain transformation will induce errors in the state variable feedback when applying the transformation.

Bandwidth considerations were also discussed and found to be similar to those specified in the continuous time formulation except for the sampling rate limitation, which is imposed to produce a stable and predictable map from the continuous time domain to the discrete time domain during the implementation process. The sampling rate for the discrete time case, however, is primarily limited for performance reasons and not because of the possibility of instability.

The experimental study in Chapter 6 sought to demonstrate the difficulties faced in underwater vehicle control and to demonstrate the capability of the methodology to successfully meet a practical control challenge. Particular emphasis was placed on illustrating the nonlinear characteristics of the vehicle motion to demonstrate the capability of the method to deal with this phenomenon in a straightforward way. The lack of a good continuous time model structure prompted the use of system identification techniques to determine a suitable discrete time model, and showed the ability of the algorithm to use this model directly.

The non-adaptive control experiments illustrated the effective trajectory following performance of the controller and the performance advantage of using the nonlinear versus the linear model. They also illustrated the ability of the algorithm to maintain stability and reasonable performance even in the presence of model uncertainty and

disturbances. The adaptive control experiments showed the large initial parameter errors that could be tolerated and the ability of the parameter estimation schemes to reduce these errors while resulting in a performance level comparable to that when the parameters were known exactly. Also revealed was the potential instability of the algorithms if special measures were not taken to guard against unmodeled dynamics. The methods proposed in Chapter 4 were found to successfully maintain stability given a reasonable estimate of the range of unmodeled dynamics present.

In reviewing what has been accomplished in this research work, several recommendations for additional or expanded work become apparent. The next logical step would be to extend the single-input results given here to the multi-input case. the presence of control inputs in more than one dimension will result in the ability to deal with nonlinear, time-variant behavior and uncertainty in more then one dimension as well. Research into the possibility of establishing a sliding mode under less than full state feedback would make the method much more general and practical to implement on systems with limited sensors.

Additional study is required into the existence of a sliding mode behavior when faced with uncertainties in system dynamics and control gain. More general stability results need to be formulated based on analytical techniques rather than the "search" strategy employed in Chapter 3.

In the adaptive controller development, an expanded stability proof to eliminate the linear boundedness condition on the system dynamics would extend its range or applicability to more general nonlinear systems. The issue of interaction with the adaptive process certainly deserves more attention in the future. Although techniques exist for dealing with the phenomenon (like those presented here), they are rather specific and usually impose assumptions or restrictions that make them less than universally applicable. It appears that since a completely general technique will probably never be found due to the diversity of system types encountered, any practical method will carry with it some special conditions to fit the situation at hand.

A key facet of any research in this area should start with a rigorous investigation of the underlying reasons for this instability mechanism. Such studies have been done in the past but usually center on a linearized analysis of system behavior. Because of the intrinsic nonlinearities present in an adaptive system, however, this approach can lead to faulty conclusions. This is because the linear analysis only reveals whether a particular equilibrium point is stable or unstable, and does not consider the nonlinear behavior that can result in stable limit cycles or other bounded behavior.

Further experimental studies should be carried out to verify the effectiveness of the methodology under different system types, disturbance environments, unmodeled dynamics, etc. Connected with these efforts should be an investigation into accommodated different system forms as well. Results from additional practical works would serve to provide insight into the existing algorithms as well as guide new research into new algorithms as required.

Considering this research work in the context of existing control methodologies and especially adaptive control, there appears to be a need for some unification and collaboration between researchers. There are many approaches existing in, for example, the discrete time literature that could be applied in the continuous time context with some modifications, and vice versa. Similarly, the linear and nonlinear system approaches could probably find some common ground between them and prove mutually beneficial to both methods.

References

1. Humphreys, D.E., and K.W. Watkinson. "Hydrodynamic Stability and Control Analysis of the UNH-EAVE Autonomous Underwater Vehicle." University of New Hampshire, Durham, 1982.
2. Lewis, D. J., and J.M. Lipscombe. "The Simulation of Remotely Operated Underwater Vehicles." *Proceedings ROV 1984, MTS*, 1984.
3. Utkin, V.I. Sliding Modes and their Application in Variable Structure Systems. Moscow: MIR, 1974.
4. Itkis, U. Control Systems of Variable Structure. New York: John Wiley and Sons, 1976.
5. Slotine, J.-J. E. "Sliding Controller Design for Nonlinear Systems." *International Journal of Control*, Vol. 40 (2): 421-434, 1984.
6. Milosavljevic, C. "General Conditions for the Existence of a Quasi- Sliding Mode on the Switching Hyperplane in Discrete Variable Structure Systems." *Automatic Remote Control*, Vol. 46: 307-314, 1985.
7. Sarpturk, S.Z., Y. Istefanopulos, and O. Kaynak. "On the Stability of Discrete Time Sliding Mode Control Systems." *IEEE Transactions on Automatic Control*, Vol. AC- 32 (10): 930-932, 1987.
8. DeLonga, D.M., J.-J.E. Slotine, and D.R. Yoerger. "Trajectory Control of Nonlinear, Time Varying, Discrete Time Systems using Discrete Time Sliding Control." Fifth International Symposium on Unmanned Untethered Submersible Technology, University of New Hampshire, Durham, 1987.
9. Jury, E.I. "On the History and Process of Sampled-Data Systems." *IEEE Control Systems Magazine*, Vol. 7 (1): 16-21, 1987.

10. Hanselmann, H. "Implementation of Digital Controllers - A Survey." Ninth IFAC World Congress on a Bridge Between Control Science and Technology, Budapest, Hungary, July, 1984.
11. Shokoohi, S., and L.M. Silverman. "Identification and Model Reduction of Time Varying Discrete Time Systems." *Automatica*, Vol. 23 (4): 509- 521, 1987.
12. Corless, M., and J. Manela. "Control of Uncertain Discrete Time Systems." *Proceedings of the American Control Conference*, Seattle, Washington, June, 1986.
13. Magaña, M.E., and Žak, S.H. "The Control of Discrete Time Uncertain Dynamic Systems." Technical Report EE - 87-32, School of Electrical Engineering, Purdue University, West Lafayette, Ind., September, 1987.
14. Åström, K. J. "Theory and Applications of Adaptive Control - A Survey." *Automatica*, Vol. 19 (5): 471-486, 1983.
15. Isermann, R. "Parameter Adaptive Control Algorithms -A Tutorial." *Automatica*, Vol. 18 (5): 513-528, 1982.
16. Landau, I.D. "A Survey of Model Reference Adaptive Techniques - Theory and Applications." *Automatica*, Vol. 10: 353-379, 1974.
17. Narendra, K.S., and L.S. Valvani. "Stable Adaptive Controller Design - Direct Control." *IEEE Transactions on Automatic Control*, Vol. AC-23 (4): 570-583, 1978.
18. Narendra, K.S., Y.-H. Lin, and L.S. Valavani. "Stable Adaptive Controller Design, Part II: Proof of Stability." *IEEE Transactions on Automatic Control*, Vol. AC-25 (3): 440-448, 1980.
19. Monopoli, R.V. "Model Reference Adaptive Control with an Augmented Error Signal." *IEEE Transactions on Automatic Control*, Vol. AC-19: 474- 484, 1974.

20. Morse, S. A. "Global Stability of Parameter-Adaptive Control Systems." *IEEE Transactions on Automatic Control*, Vol. AC-25 (3): 433- 439, 1980.
21. Narendra, K. S., and L.S. Valavani. "Discrete and Indirect Model Reference Adaptive Control." *Automatica*, Vol. 15: 653-664, 1979.
22. Middleton, R.H., and G.C. Goodwin. "Adaptive Control of Time Varying Linear Systems." *IEEE Transactions on Automatic Control*, Vol. 33 (2): 150-155, 1988.
23. Narendra, K.S., and A.M. Annaswamy. "A New Adaptive Law for Robust Adaptation Without Persistent Excitation." *IEEE Transactions on Automatic Control*, Vol. AC-32 (2): 134-145, 1987.
24. Lee, T.-H., and K.S. Narendra. "Stable Discrete Adaptive Control with Unknown High-Frequency Gain." *IEEE Transaction. on Automatic Control*, Vol. AC-31 (5): 477-479, 1986.
25. Kreisselmeier, G., and M.C. Smith. "Stable Adaptive Regulation of Arbitrary n th-Order Plants." *IEEE Transaction on Automatic Control*, Vol. AC- 31 (4): 299-305, 1986.
26. Ioannou, P.A., and K.S. Tsakalis. "A Robust Direct Adaptive Controller." *IEEE Transactions on Automatic Control*, Vol. AC-31 (11): 1033-1043, 1986.
27. Ioannou, P.A. "Robust Adaptive Controller with Zero Residual Tracking Errors." *IEEE Transactions on Automatic Control*, Vol. AC-31 (8): 773- 776, 1986.
28. Goodwin, G.C., D.J. Hill, and M. Palaniswami. "A Perspective on Convergence of Adaptive Control Algorithms." *Automatica*, Vol. 20 (5): 519-531, 1984.
29. Xianya, X., and R.J. Evans. "Discrete Time Adaptive Control for Deterministic Time Varying Systems." *Automatica*, Vol. 20 (3): 309-319, 1984.
30. Slotine, J.-J.E., and W. Li. "Adaptive Manipulator Control - A Case Study." *IEEE International Conference on Robotics and Automation*, Raleigh, NC, 1987.

31. Slotine, J.-J.E., and W. Li. "Adaptive Robot Control: A New Perspective." IEEE Conference on Decision and Control, Los Angeles, 1987.
32. Slotine, J.-J.E. and W. Li. "Theoretical Issues in Adaptive Manipulator Control." Fifth Yale Workshop on Applications of Adaptive Systems Theory, 1987.
33. Tsakalis, K., and P.A. Ioannou. "Adaptive Control of Linear Time Varying Plants." *Automatica*, Vol. 23 (4): 459-468, 1987.
34. Balestrino, A., G. De Maria, and A.S.I. Zinober. "Nonlinear Adaptive Model-Following Control." *Automatica*, Vol. 20 (5): 559-568, 1984.
35. Goodwin, G.C., and D.Q. Mayne. "A Parameter Estimation Perspective of Continuous Time Model Reference Adaptive Control." *Automatica*, Vol. 23 (1): 57-70, 1987.
36. Praly, L. "Robustness of Model Reference Adaptive Control." Third Yale Workshop on Applications of Adaptive System Theory, 1983.
37. Mahmoud, M.S., H.A. Othman, and N.M. Khraishi. "Reduced-Order Performance of Adaptive Control Systems." *IEEE Transactions on Automatic Control*, Vol. AC-31 (11): 1076-1079, 1986.
38. Rohrs, C.E., L.S. Valavani, and M. Athans. "Robustness of Adaptive Control Algorithms in the Presence of Unmodeled Dynamics." 21st IEEE Conference on Decision and Control, Orlando, FL, 1982.
39. Ioannou, P.A., and P.V. Kokotovic. "Singular Perturbations and Robust Redesign of Adaptive Control." 21st IEEE Conference on Decision and Control, Orlando, FL, 1982.
40. Kosut, R.L., and B. Friedlander. "Performance Robustness Properties of Adaptive Control Systems." 21st IEEE Conference on Decision and Control, Orlando, FL, 1982.

41. Johnson, C.R., and G.C. Goodwin. "Robustness Issues in Adaptive Control." 21st IEEE Conference on Decision and Control, Orlando, FL, 1982.
42. Riedle, B.D., and P.V. Kokotovic. "A Stability-Instability Boundary for Disturbance-Free Slow Adaptation with Unmodeled Dynamics." *IEEE Transactions on Automatic Control*, Vol. AC-30 (10): 1027-1030, 1985.
43. Ioannou, P.A., and P.V. Kokotovic. "Instability Analysis and Improvement of Robustness of Adaptive Control." *Automatica*, Vol. 20 (5): 583-594, 1984.
44. Krause, J., M. Athans, S. Sastry, and L.S. Valvani. "Robustness Studies in Adaptive Control." 22nd IEEE Conference on Decision and Control, San Antonio, TX, 1983.
45. Anderson, B.D.O., "Adaptive Systems, Lack of Persistency of Excitation, and Bursting Phenomena." *Automatica*, Vol. 21 (3): 247-258, 1985.
46. Kosut, R.L., R.C. Johnson, and B.D.O. Anderson. "Robustness of Reduced-Order Adaptive Model Following." Third Yale Workshop on Applications of Adaptive Systems Theory, 1983.
47. Kokotovic, P.V., R.E. O'Malley, and P. Sanuti. "Singular Perturbations and Order Reduction in Control Theory - An Overview." *Automatica*, Vol. 12: 123-132, 1976.
48. Hsu, L. and R.R. Costa. "Bursting Phenomena in Continuous Time Adaptive Systems with a σ -Modification." *IEEE Transactions on Automatic Control*, Vol. AC-32 (1): 84-86, 1987.
49. Narendra, K.S. and A.M. Annaswamy. "A General Approach to the Stability Analysis of Adaptive Systems." 23rd IEEE Conference on Decision and Control, Las Vegas, NV, 1984.

50. Åström, K. J., "Interactions Between Excitation and Unmodeled Dynamics in Adaptive Control." 23rd IEEE Conference on Decision and Control, Las Vegas, NV, 1984.
51. Slotine, J.-J.E. and J.A. Coetsee. "Adaptive Sliding Controller Synthesis for Nonlinear Systems." *International Journal of Control*, Vol 42 (6), 1986.
52. Kreisselmeier, G., and B.D.O. Anderson. "Robust Model Reference Adaptive Control." *IEEE Transactions on Automatic Control*, Vol. AC-31 (2): 127- 133, 1986.
53. Peterson, B.P. and K.S. Narendra. "Bounded Error Adaptive Control." *IEEE Transactions on Automatic Control*, Vol. AC-27 (6): 1161-1186, 1982.
54. Orlicki, D., L.S. Valavani, M. Athans, and G. Stein. "Adaptive Control with Variable Dead-Zone Nonlinearities." American Control Conference, San Diego, 1984.
55. Anderson, B.D.O., R.R. Bitmead, C.R. Johnson, P.V. Kokotivic, R.L. Kosut, I.M.Y. Mareels, L. Praly, B.D. Riedle. Stability of Adaptive Systems. Cambridge, MA: MIT Press, 1986.
56. Kosut, R.L., B.D.O. Anderson, and I.M.Y. Mareels. "Stability Theory for Adaptive Systems: Method of Averaging and Persistency of Excitation," *IEEE Transactions on Automatic Control*, Vol. AC -32 (1): 26-34, 1987.
57. Narendra, K.S., and A.M. Annaswamy. "Robust Adaptive Control in the Presence of Bounded Disturbances." *IEEE Transactions on Automatic Control*, Vol. AC-31 (4): 306-315, 1986.
58. Narendra, K.S., and A.M. Annaswamy. "Persistent Excitation and Robust Adaptive Algorithms." Third Yale Workshop on Applications of Adaptive Systems Theory, 1983.

59. Anderson, B.D.O., and R.C. Johnson. "Exponential Convergence of Adaptive Identification and Control Algorithms." *Automatica*, Vol. 18 (1): 1-13, 1982.
60. Bodson, M. and S. Sastry. "Exponential Convergence and Robustness Margins in Adaptive Control." 23rd IEEE Conference on Decision and Control, Las Vegas, NV, 1984.
61. Anderson, B.D.O. "Exponential Convergence and Persistent Excitation." 21st IEEE Conference on Decision and Control, Orlando, FL, 1982.
62. Wittenmark, B., and K.J. Åström. "Practical Issues in the Implementation of Self-Tuning Control." *Automatica*, Vol. 20 (5): 595- 605, 1984.
63. Rohrs, C.E., M. Athans, L.S. Valavani, and G. Stein. "Some Design Guidelines for Discrete Time Adaptive Controllers." *Automatica*, Vol. 20 (5): 653-660, 1984.
64. Middleton, R.H., G.C. Goodwin, D.J. Hill, and D.Q. Mayne. "Design Issues in Adaptive Control." *IEEE Transactions on Automatic Control*, Vol. 33 (1): 50-58, 1988.
65. Bristol, E.H. "Experimental Analysis of Adaptive Controllers." Third Yale Workshop on Applications of Adaptive Systems Theory, 1983.
66. Yoerger, D.R., and Slotine, J.-J. E. "Robust Trajectory Control of Underwater Vehicles." *IEEE Journal of Ocean Engineering*, Vol. OE-10 (4), 1985.
67. Yoerger, D.R., and J.B. Newman. "Supervisory Control of Underwater Vehicles and Manipulators." *Proceedings ROV 1985, MTS*, 1985.
68. Franz, P., and D. Limbert. "A Thrust Vector Controller for the EAVE Vehicle." Fifth International Symposium on Unmanned, Untethered Submersible Technology, University of New Hampshire, Durham, 1987.
69. Luenberger, D.G. Introduction to Dynamic Systems. New York: John Wiley and Sons, 1979.

70. Ogata, K. Modern Control Engineering. Englewood Cliffs, NJ: Prentice Hall, Inc., 1970.
71. Åström, K.J., and B. Wittenmark. Computer Controller Systems. Englewood Cliffs, NJ: Prentice Hall, Inc., 1984.
72. Goodwin, G.C., and K.S. Sin. Adaptive Filtering, Prediction, and Control. Englewood Cliffs, NJ: Prentice Hall, Inc., 1984.
73. Armstrong, B. "On Finding 'Exciting' Trajectories for Identification Experiments Involving Systems with Nonlinear Dynamics." *IEEE International Conference on Robotics and Automation*, Raleigh, N.C., 1987.
74. Li, W. and J.-J.E. Slotine. "Parameter Estimation Strategies for Robotic Applications." ASME Winter Annual Meeting, Boston, 1987.
75. Boyd, S. and S. Sastry. "Necessary and Sufficient Conditions for Parameter Convergence in Adaptive Control." *Automatica*, Vol. 22 (6): 629-639, 1986.
76. Cristi, R. "Adaptive Control with Finite Time Persistency of Excitation." *IEEE Transactions on Automatic Control*, Vol. AC-31 (10): 982-985, 1986.
77. Krause, J.M. and P. Khargonekar. "Parameter Information Content of Measurable Signals in Direct Adaptive Control." *IEEE Transactions on Automatic Control*, Vol. AC-32 (9): 802-810, 1987.
78. Åström, K.J. and P. Eykhoff. "System Identification - A Survey." *Automatica*, Vol. 7: 123-162, 1971.
79. Sandman, B.E. and J.G. Kelly. "System Identification: Application to Underwater Vehicle Dynamics." *Journal of Hydronautics*, Vol. 8 (3): 94- 99, 1974.
80. Smith, J.E. Mathematical Modeling and Digital Simulation for Engineers and Scientists. New York: John Wiley and Sons, Inc., 1977.

81. Casti, L. Nonlinear System Theory. Orlando, FL: Academic Press, Inc., 1985.
82. Žak, S.H. and C.A. MacCarley. "State-Feedback Control of Non-Linear Systems." *International Journal of Control*, Vol. 43 (5): 1497- 1514, 1986.
83. Lee, H.-G. and S.I. Marcus. "Appropriate and Local Linearizability of Non-Linear Discrete-Time Systems." *International Journal of Control*, Vol. 44 (4): 1103-1124, 1986.
84. Hunt, L.R., R. Su, and G. Meyer. "Global Transformations of Nonlinear Systems." *IEEE Transactions on Automatic Control*, Vol. AC-28 (1): 24- 31, 1983.
85. Krener, A.J. and A. Isidori. "Linearization by Output Injection and Nonlinear Observers." *Systems and Control Letters*, Vol. 3 (1): 47-52, 1983.
86. Asada, H. and J.-J. E. Slotine. Robot Analysis and Control. New York: John Wiley and Sons, Inc., 1986.
87. Abkowitz, M.A. Stability and Motion Control of Ocean Vehicles. Cambridge: MIT Press, 1969.
88. Gertler, M. and G.R. Hagen. "Standard Equations of Motion for Submarine Simulation." *Research and Development Report 2510*, Naval Ship Research and Development Center, Washington, DC, June, 1967.